Documentation

HAFAS Raw data file format

Input files for data transformation

Version 5.20.39

HaCon Ing. GmbH, 18th May 2010

0. Contents

1. History of changes

Date	Change
10/01/2001	Version 4.16.1; Author: Thomas Müller Addition of the description in **GR*-lines in f-plan files (see Section **GR*-line (optional):
31/01/2001	Version 4.16.2; Author: Fiekert In the section Operator information [BETRIEB] the old 4-digit format have been replaced by the new 5-digit format
19/09/2001	Version 4.20.3 (only version number changed)
05/03/2002	General improvement. Author: Burre
07/04/2003	Version 5.00.6; Author: Lücke Removed the region file
15/07/2003	Version 5.00.8 Author: Fiekert Corrected type for KW-numbers
05/05/2004	Version 5.00.9 Author: Fiekert Corrected description of file VEREINIG
21.05.2004	Version 5.00.10, Author: Frank Geveke Additional information on interchanges between stops
25.10.2004	Version 5.20.3, Author: Frank Geveke New section 5.3.12 about check-in and check-out times
26.01.2005	Version 5.20.4, Author: Frank Geveke New section 6.21 about the file HAUSNR. Z coordinate added in the file BFKOORD (section 5.2)
18.02.2005	Version 5.20.5, Author: Frank Geveke New coordinate format for file SPERRKANTEN
24.03.2005	Version 5.20.6, Author: Wolfram Fiekert Additional restrictions in BHFART for individual routing
30.03.2005	Version 5.20.7, Author: Thomas Schwartz Extended definition of stop groups
12.04.2005	Version 5.20.8, Author: Wolfram Fiekert *B-line for METABHF; formatting
14.04.2005	Version 5.20.9, Author: Thomas Müller section 5.5 [ZUGART] revised
18.07.2005	Version 5.20.10, Author: Frank Geveke Hidden links
24.10.2005	Version 5.20.11, Author: Thomas Schwartz New: guaranteed interchanges, guaranteed walking links, distances of walking links

continuation on next page

continuation of previous page

Date	Change
19.01.2006	Version 5.20.12, Author: Thomas Müller, Wolfram Fiekert Cosmetical changes, adaptions of the examples of the version for 9 digit stop numbers
27.01.2006	Version 5.20.13, Author: Frank Geveke Added crossings in file HAUSNR (section 6.21). Additional specification of *-lines in file METABHF (sections 5.6) which can be used with walking links connecting different pools.
05.05.2006	Version 5.20.14, Author: Frank Geveke The duration of a walking link can be defined in minutes and seconds. New T-Line for file bhfart (section 6.1) defines allowd types of individual traffic for the routing between locations and stops.
10.05.2006	Version 5.20.15, Author: Thomas Müller Example for *T-services corrected
02.08.2006	Version 5.20.16, Author: Thomas Schwartz Language definitions for stop names (file BAHNHOF)
29.08.2006	Version 5.20.17, Author: Frank Geveke H-equivalence
20.11.2006	Version 5.20.18, Author: Wolfram Fiekert Extended definition of continuous operations
07.12.2006	Version 5.20.19, Author: Robin Giese Line number 8-digit
15.12.2006	Version 5.20.20, Author: Wolfram Fiekert Correction of the describing tabular of UMSTEIGZ
06.07.2007	Version 5.20.21, Author: Tobias Ehbrecht Introduction of meta attributes
10.09.2007	Version 5.20.21, Author: Tobias Ehbrecht Added extended infotexts
09.10.2007	Version 5.20.21, Author: Tobias Ehbrecht Fixed some minor mistakes
26.10.2007	Version 5.20.22, Author: Lars Dietzel Realgraph added
04.12.2007	Version 5.20.23, Author: Robin Giese Attributes with infotext
17.12.2007	Version 5.20.24, Author: Lars Dietzel Regions and hailed shared taxis added

continuation on next page

continuation of previous page

Date	Change
08.01.2007	Version 5.20.24, Author: Lars Dietzel Corrected typing errors, removed bad reference
10.01.2008	Version 5.20.25, Author: Lars Dietzel Restricted service types for realgraph edges
23.01.2008	Version 5.20.26, Author: Lars Dietzel Changed order of service number/administration in service mapping data
04.03.2008	Version 5.20.27, Author: Tobias Ehbrecht Added through coach blocks
27.03.2008	Version 5.20.27, Author: Lars Dietzel Corrected extended infotexts example
15.05.2008	Version 5.20.28, Author: Wolfram Fiekert Lengths of direction and border texts adopted
26.08.2008	Version 5.20.29, Author: Lars Dietzel Infotexts and flags for realgraph edges
27.08.2008	Version 5.20.30, Author: Lars Dietzel Infotexts and flags for realgraph nodes
04.09.2008	Version 5.20.30, Author: Lars Dietzel Minor changes
09.12.2008	Version 5.20.30, Author: Fabian Leitritz Commonly valid infotexts
04.06.2009	Version 5.20.31, Author: Fabian Leitritz Processing sequence of interchange times in *Hafas*
19.06.2009	Version 5.20.32, Author: Lars Dietzel Corrected default name for service-route-network-mapping file
29.07.2009	Version 5.20.33, Author: Lars Dietzel File fahrtzuordng, **P*-lines: some clarifications
06.08.2009	Version 5.20.34, Author: Thomas Schwartz Type definition for station names
09.10.2009	Version 5.20.35, Autor: Lars Dietzel Added realgraph for footpaths, added footpath numbers
23.11.2009	Version 5.20.36, Autor: Fabian Leitritz Added **E*-lines (sections with no interchanging, ...)
18.01.2010	Version 5.20.36, Autor: Fabian Leitritz Corrected column width within the time displacement file

continuation on next page

continuation of previous page

Date	Change
09.04.2010	Version 5.20.37, Autor: Fabian Leitritz Minor fixes
18.05.2010	Version 5.20.38, Autor: Lars Dietzel Missing constraint for operating days for services added, clarification on processing of file fahrtzuordng
18.05.2010	Version 5.20.39, Autor: Lars Dietzel UIC country codes revised, documentation for service type pictures completed, typos corrected

2. Introduction

The timetable data of a transport provider must be available in the *Hafas* raw data format in order to obtain advantange of the *Hafas* timetable information system. This format is the starting point for the *Transform* data treatment which generates a specialised binary data format from the raw data. This binary data format is tailored to the needs of the search algorithm and thus enables optimum information to be found in a very short time. Special features of the *Hafas* raw data format include the following:

- not dependent on a specific operating system since all data files are plain text files,
- editable since it is largely to be used as a readable document,
- flexible since even complicated facts can be represented,
- not database dependent, yet easily imported or exported by popular database systems.

3. Name conventions

To make this description comprehensible, the following name conventions apply:

- The term “service” refers to different means of transport such as trains, buses, ships, etc.
- The term “stop” is correspondingly used to represent a railway station, tram stop, bus terminal, etc.
- A “route” is a sequence of stops accomplished by a service. The stops are arranged in the order in which they are passed through.

4. Data set

First of all we would like to present an overview of the required data. All data are grouped in individual files. The file name is given in the heading of the appropriate chapter. The name is placed in square brackets and written upper case. The data are divided into two groups:

1. Mandatory data,
2. Optional data.

Mandatory data are:

- the list of all stops (the stop index),
- coordinates of stops,
- the timetable data (services with arrival/departure times),
- information on the operating days of the services,
- range of timetable period,
- information on the means of transport,
- footpaths between stops and grouping of stops.

Optional data are important for the *Hafas* information system:

- list of interchange stops with ranking of their importance,
- priorities for selecting a stop as an interchange stop in case several interchange stops are possible,
- additional attributes for services,
- definition of interchange times (service related interchange times, line/direction related interchange times, interchange times related to means of transport),
- through services (two separately listed services are in reality one service),
- couplage of portions of trains,
- direction related information,

- address and building data,
- information on platforms, bus shelters or similar,
- the allocation of stops to regions.

To ensure that these data are independent from operating systems, all data files are written as text files in the IBM PC character set (8 bits). Conversion into a simple ASCII character set (7 bits) is possible. Country-specific characters (umlauts, accents) can be used as far as they are contained in the IBM character set, otherwise they should be simplified.

4.1. Version specific limitations

This documentation is a complete description of all possibilities within the *Hafas* raw data format. Some possibilities might not be available to all implementations of *Hafas*/*Transform* either due to older program versions or due to possibilities not licenced within a certain customer.

5. Mandatory data files

5.1. The stop index [BAHNHOF]

The stop index is a complete list of all stops occurring in the entire timetable data of an organisation. The stop file contains for each stop:

- a unique 7-digit-digit number,
- abbreviation for a transport authority and
- the stop name.

The unique 7-digit stop number consists of the UIC country code which occupies the first numerals, and of a further five numerals of your choice (with the exception of the sequence 00000). The exchange of data comprising different countries or transport authorities essentially depends on harmonisation of the existing stop database. At present, the following UIC numbers are valid:

10 Finland
20 Russia
21 Belorussia
22 Ukraine
23 Moldova
24 Lithuania
25 Latvia
26 Estonia
27 Kazakhstan
28 Georgia
29 Uzbekistan
30 North Korea
31 Mongolia
32 Vietnam
33 China
40 Cuba
41 Albania
42 Japan
44 Bosnia-Herzegovina (Serbian part)
50 Bosnia-Herzegovina (Bosnian-Croatian part)
51 Poland
52 Bulgaria
53 Rumania
54 Czech Republic
55 Hungary
56 Slovakia
57 Azerbaijan
58 Armenia
59 Kyrgyzstan
60 Ireland
61 South Korea
62 Montenegro
65 Macedonia
66 Tadzhikistan
67 Turkmenistan
70 Great Britain
71 Spain
72 Serbia
73 Greece
74 Sweden
75 Turkey
76 Norway
78 Croatia
79 Slovenia
80 Germany
81 Austria
82 Luxembourg
83 Italy
84 Netherlands
85 Switzerland
86 Denmark
87 France
88 Belgium
89 Bosnia
90 Egypt
91 Tunisia
92 Algeria
93 Morocco
94 Portugal
95 Israel
96 Iran
97 Syria
98 Lebanon
99 Iraq

The initial digits 01 - 09 not used in the UIC code are reserved to provide the option of including local transport authorities.

The stops file contains one entry per line in the following form:[1]

File format:

Column	Type	Meaning
1−7	*INT32*	The number of the stop
9−11	*CHAR*	Blank (3 blanks) or 3 characters for transport authority information
13−62	*CHAR*	Name(s) of the stop, including optional language definitions

- Names may contain blanks.
- A maximum of the first **50** characters of a name is accepted.
- Several names are separated by $, e.g. Geneva$Genf.
- If several names are given, the first name will serve as default name for the system output.
- Each station may have different names corresponding to different languages. One oder more languages are assigned to a station name by inserting a pattern like `<language code1...language codeN>`, which follows the corresponding name. A $ character separates the name from its language definition (see example below).
- A further method to select different names for the same station is to assign special types to the station names. These abstract types have their values within the range from 1 to 9. The syntax of a type definition is similar to a language definition. Language and type definitions may be mixed (see example below).

Possible language codes are as follows:

d German

e English

f French

[1] For the meaning of the type name, see explanations in chapter 7

- `h` Spanish
- `i` Italian
- `j` Japanese
- `n` Dutch
- `o` Norwegian
- `p` Polish
- `s` Swedish
- `t` Turkish
- `u` Hungarian
- `w` Danish
- `x` Irish

Example:

```
8800002     Bruxelles-Nord$Brüssel Nord$<d>$Brussel Noord$<n>
8833001     Louvain [B]$<f>$Leuven [B]$<n>$Louvain$<f2>Louv$<f1>
1234567     AAAA$<d12><f23>$BBBB$<e1><f1>
7654321     CCCC$<12>$DDDD$<34>
```

Explanation:

The default name of the stop 8800002 is ”Bruxelles-Nord”’. ”Brüssel-Nord”’ is an alternative name, which is in addition defined to be a german name. ”Brussel Noord”’ is another alternative name, but a dutch one. Alternative names are used e.g. for selecting stops.

The default name of the stop 8833001 is ”‘Louvain [B]”’, which is a french name. The dutch name is ”‘Leuven [B]”’. Two other french names are listed, but these have special types: ”‘Louv”’ has type 1 and ”‘Louvain”’ has type 2.

Stop 1234567 has two names. The first is ”‘AAAA”’, which is a german name, carrying type 1 and 2. Moreover, this is a french name, having types 2 and 3. The second name reads as ”‘BBBB”’, which is used in English and French and it has type 1.

Stop 7654321 has two names, which are used in any language, but have different types. The first name reads as "'CCCC"', having types 1 and 2. The second reads as "'DDDD"' and has types 3 and 4.

5.2. The coordinates of stops [BFKOORD]

The stop coordinates are used by *Hafas* for visualisation of the passage of a journey, for checking the journey times and for optimising the search algorithm. The following is stored for each stop:

- the stop number,
- the X/Y coordinates,
- die Z coordinate (optional)
- the stop name.

File format:

Column	Type	Meaning
$1-7$	*INT32*	Number of the stop
$9-18$	*FLOAT*	X-Coordinate
$20-29$	*FLOAT*	Y-Coordinate
$31-36$	*INT16*	Z-Coordinate (optional)
$38ff$	*CHAR*	Comment character “%” followed by plain text of the stop name (optional for better readability)

The X/Y coordinates are indicated by degrees of longitude and latitude in a geographical coordinate system. Units: degrees with decimal places. For accurancy in meters it has to have 6 right-of-comma positions. The preferred coordinate system is WGS84.

Alternatively coordinates are stated in kilometres in a rectangular reference system with selectable zero point. During the data preparation the program MAKEGEO must be indicated by switch which coordinate system is to be used (default are geographical coordinates). The switch -p changes to the rectangular coordinate system.

In opposition to X- and Y-coordinates the Z-coordinate must be the height above sea level given in meters.

Example:

```
8000261 11.5582710 48.1402880      % München Hbf
8000262 11.6049710 48.1274400 450  % München Ost
```

Explanation:

The stop “Munich Hbf” has the number 8000261, lies on the degree of longitude 11.558271 and the degree of latitude 48.140288 (says WGS84). The stop “Munich Ost” has the number 8000262, lies on the degree of longitude 11.604971 and the degree of latitude 48.12744 (says WGS84) and lies 450 meters above sea level.

5.3. The timetable file(s) [FPLAN]

The timetable files contain the data for all services. In each case they contain the complete service descriptions with:

- a code for each service (service number, administration),
- the means of transport or train categories,
- the days of operation for the service,
- other servicing attributes such as bicycle conveyance etc.,
- the stops passed through with the corresponding arrival and departure times arranged in order of advancing time,
- optional information on line name and direction of service,
- optional information on the route of the journey (crossing of national borders, tariff boundaries, etc. ...).

5.3.1. Service description

For the complete description of a service, two kinds of data lines are used:

- lines which define the service number, the operating days of the service, the service category and similar. These lines begin with a *.
- data lines which describe the route of the service, i.e. the stops served with the arrival and departure times. The arrival and departure times contain 6 characters (5 digits and sign). The previous format with 4 digits is still usable until further notice (backward compatibility).

The lines can be mixed, but the lines which describe the route of service have to be chronologically ascending. All lines are organized the way, that the character “%” (for comments) **has to** appear on position column 59. After that, any characters of your choice are allowed for the remainder of the line.

If only times in 4 digits are used, the position of the character “%” ist column 55. It’s not allowed to mix up times in 4 and in 6 digits.

For representation of the service, the following lines are essential:

1. **Z*-line, **KW*-line or **T*-line with the service number. This line introduces the start of a new service,
2. **G*-line to define the means of transport,
3. **A VE*-lines to define the operating days of the service.

Remark:

There’s only one **Z*-, **KW* or **T*-line in each case. All other *-lines may appear several times per service, if necessary. In *-lines departure stations are always searched from the beginning of the route (first appearence of stop number). Final stations, which are only identified by the number of the stop, are searched from the last place in the route (last appearence of stop number).

If a characteristic shall apply from the departure station, the stop number of a *-line can be left out. This is also valid for the number of the arrival station. In this case the *-line applies up to the end of service.

There are the following possibilities to define the validity area of *-lines in the route despite more than once appearing stop numbers:

Example:

```
*Z 01554 80____                                                %
*G ICE 8010085 8010097                                         %
*A VE 8010085 8010097 046149                                   %
*A BR      #2 8010097                                          %
*A BW 8010085 8010097                  1857                    %
*A L  8010097 8010097            #0     #1                     %
*A LS      #6 8010097            #1   2200                     %
8010085 Dresden Hbf                    1611                    %
8010205 Leipzig Hbf             1718   1722                    %
8010366 Weimar                  1814   1815                    %
8010101 Erfurt Hbf              1828   1830                    %
8010097 Eisenach                1857   1858                    %
8000115 Fulda                   1943   1945                    %
8000105 Frankfurt(Main)Hbf      2036   2040                    %
8000152 Hannover Hbf            2050   2100                    %
8010097 Eisenach                2200                           %
```

Explanation:

Route index instead of stop number

It is possible to use an index on a certain route line instead of a stop number. The index count starts with 0.

```
*A BR      #2 8010097                                          %
```

The attribute applies starting with the 3rd station in the route (Weimar). The remaining space in this column must be filled with space characters.

Setting of arrival or departure times

- To define the start of the appliance the departure time of the choosen stop has to be set. Time and stop number must match to the route. The route is searched through from the beginning.
- To define the end of the appliance the arrival time of the choosen stop has to be set. Time and stop number must match to the route. The route is searched through from the end.

```
*A BW 8010085 8010097                    1857                    %
```

The attribute applies up to the stop with the according arrival time (18:57) and the stop number 8010097.

Index on the nth occurence of a stop number within the route

The nth occurence (count from 0) of the stop number within the route is set as validity start or stop.

```
*A L  8010097 8010097            #0     #1                      %
```

Remark:

If an index is set on the route (instead of the stop number), the according later indication of time or index is irrelevant and will be ignored.

You'll have to consider the width of the actual columns (stop number or times), if an index or the time is set. Places have to be filled with blanks, if necessary.

This also applies, if no stop number is set (empty field).

```
*A LS      #6 8010097            #1   2200                    %
```

- **A*-line
- Attribute is LS
- The validity start in the route is the 7th stop (Frankfurt(Main) Hbf).
- Valid to Eisenach (8010097)
- This attribute would be valid from the 2nd occurence of this departure station, if there would not have been set an index on the route (#6).
- Validity end is the last found stop in the route with the number 8010097 and the arrival time 22:00.

5.3.2. *Z-lines:

The data of a new service begin with the *Z-line. In order to be able to identify a service, the service number and the so-called administration number are stated in the *Z-line. As a result, it is thus possible to differentiate between services with the same number.

File format:

Column	Type	Meaning
1−2	*CHAR*	*Z
4−8	*INT32*	Service number
10−15	*CHAR*	Administration (6 digits); The administration entry must not contain any blanks.
17−21	*INT16*	Empty
23−25	*INT16*	(Optional) number of intervals; specifies the number of intervals to follow.
27−29	*INT16*	(Optional) time offset in minutes (between two services).

Example:

```
*Z 01554 80____       023 010                                  %
```

Explanation:

- Line type = *Z
- Travel number (extern train number) = 01554
- Administration number = 80____
- Number of intervals = 23
- Time between the intervals = 10 minutes

5.3.3. *T-lines:

The data of a new service begin with *T-lines (*T-Service). To identify a service, the service and the so-called administration number are indicated in the *T-line. So services with the same number can be differentiated. The service numbers should be differentiated from those by services with *Z-lines.

Contrary to *Z-services the times of *T-services are regarded as “indistinct”. This means that with *T-services traffic is illustrated, which’s clock density are known, but the actual departure-/arrivaltimes are unknown (or at least uncertainly).

The route-lines of a *T-service indicate the times of the theoretical earliest service. The service frequency is indicated by a time interval (in minutes), within those services start at first stop of the service after the indicated point of starting time. From the times of the indicated first service and the time interval in the *T-line the period results, within whose the appropriate service operates.

The clock densities are indicated in seconds, in order to be able to indicate to the User e.g. a 7.5 minute pulse correctly. It is counted with on minutes rounded up times internally.

With each entrance at the beginning of a service the starting point of the search plus the time between intervals is used as entrance time into a *T-service. At transfers into a *T-service the arrival time at the stop plus the regular transfer time (the same rules apply as to *Z-services) plus a clock duration are used as entrance time (technical *T-services operate in minute pulse with one for the cycle time increased transfer time) . *T-services must not be used with through connections and links. In addition *T-services are not summarized automatically concerning their traffic days or intervals.

File format:

Column	Type	Meaning
1−2	*CHAR*	*T
4−8	*INT32*	Service number
10−15	*CHAR*	Administration (6 digits); There must not be any blanks
17−20	*INT16*	Running time in minutes
22−25	*INT16*	Clock density in seconds (Distance between 2 services).

Example:

```
*T 01554 80____ 0240 0450                                    %
```

Explanation:

- Line type = **T*
- Service number (external train number) = 01554
- Administration number = 80____
- If the beginning of the search is with 8:00 o'clock (depending on the following, here not specified route lines), connections till 12:00 o'clock (4 hours, according to 240 minutes) will be showed.
- Clock density = 7.5 minutes, according to 450 seconds. This clock density is used for increasing the transfer time.

5.3.4. *G-lines:

These lines are used to define the means of transport (train, bus, tram, etc.) or the service category (ICE, IC, IR, etc.) for each journey section. If required, several lines can be specified. The information must cover the route completely and unambiguously. The encoding of the service category is set in file ZUGART. Each means of transport, respectively each service category must be included in this file.

File format:

Column	Type	Meaning
1−2	*CHAR*	**G*
4−6	*CHAR*	mean of transport or service category
8−14	*[#]INT32*	(optional) stop number from which the definition is set.
16−22	*[#]INT32*	(optional)stop number up to which the definition is set.
24−29	*[#]INT32*	(optional) Index for the xth occurrence or point of departure time
31−36	*[#]INT32*	(optional) Index for the xth occurrence or point of arrival time

Example:

```
*G ICE 8010085 8010097                                          %
```

Explanation:

- Line type = **G*
- Mean of transport = ICE
- valid from stop 8010085 to stop 8010097

5.3.5. **A VE*-lines:

The **A VE*-lines indicate the operating days of a service. The operating day information can apply to the entire route or change section by section. The operating day information must cover the route completely and unambiguously. For each day there must be at most one operated route section, that is to say that there must not be “gaps” in the route of the service. The operating day number “**000000**” indicates “daily service”; for all other numbers there are corresponding entries in a separate file (see file BITFELD).

Remark:

To increase performance it is important for the timetable information system that the operating day number “000000” (and only this) is used to denote “daily service”.

File format:

Column	Type	Meaning
1−5	*CHAR*	**A VE*
7−13	*[#]INT32*	(optional) Index or stop number from which the operating days apply in the route.
15−21	*[#]INT32*	(optional) Index or stop number up to which the operating days apply in the route.
23−28	*INT32*	(optional) operating day number for days on which the service takes place. If this information is missing, this service runs daily (equivalent to 000000).
30−35	*[#]INT32*	(optional) Index for the xth occurrence or point of departure time
37−42	*[#]INT32*	(optional) Index for the xth occurrence or point of arrival time

Example:

```
*A VE 8010085 8010097 046149                                    %
```

Explanation:

- Line type = **A VE*
- Valid from stop 8010085 to stop 8010097
- Applies on the traffic days 046149 (etry in the file BITFELD)

5.3.6. *A-lines (optional)

The attribute lines serve to assign additional information to the service. These can be notes on catering facilities for the train, notes on reservation, sleepers, etc. The definition of attributes used (attribute code) is performed in a special file (see file ATTRIBUT).

File format:

Column	Type	Meaning
1−2	*CHAR*	**A*
4−5	*CHAR*	attribute code
7−13	*[#]INT32*	(optional) (optional) Index or stop number from which the attribute applies in the route.
15−21	*[#]INT32*	(optional) (optional) Index or stop number up to which the attribute applies in the route.
23−28	*INT16*	(optional) Bitfieldnumber for the days, when the attribute applies. Is this Indication missing, the attribute applies anytime.
30−35	*INT16*	(optional) Index for the xth occurrence or point of departure time
37−42	*INT16*	(optional) Index for the xth occurrence or point of arrival time

Example:

```
*A BB 8010366 8010097                                    %
```

Explanation:

- Line type = **A*
- Attribute code = BB
- Valid from stop 8010366 to stop 9010097
- Applies daily, because there are no restrictions.

The following are examples of attribute definitions (file ATTRIBUT):

“BB” bus does not stop at the railway station

“X” stop on request

“L” train running with couchettes only

5.3.7. *I-lines (optional):

On the basis of the information text lines it is possible to supplement a journey with additional information in plain text. Whilst a particular text is permanently assigned to an attribute code in the case of train attributes (see *A) it is possible – depending on the journey involved - to indicate various texts in relation to an infotext code.

File format:

Column	Type	Meaning
1−2	*CHAR*	*I
4−5	*CHAR*	Infotext code. If "'XI"' is used, the infotext is treated as if containing XML.
7−13	*[#]INT32*	(optional) Index or stop number from which the infotext applies in the route.
15−21	*[#]INT32*	(optional) Index or stop number to which the infotext applies in the route.
23−28	*INT16*	(optional) Bitfieldnumber for the days, when the infotext applies. Is this Indication missing, the infotext applies anytime.
30−36	*INT32*	(optional) Number of the information text from the file INFOTEXT.
38−43	*INT16*	(optional) Index for the xth occurrence or point of departure time
45−50	*INT16*	(optional) Index for the xth occurrence or point of arrival time

The infotext code stands for a group of info texts, e.g. information about the operator or the train. These groups are read in *Hafas* and showed according to the defaults (e.g. infotexts are only showed on printed media and not with the timetable information itself).

In the raw data is to be paid attention for that not 2 equal infotext codes with different meaning exist.

Example:

```
*I ZN 8010366 8010097         0002905                    %
```

Explanation:

- Line type = *I*
- Infotext code = ZN
- Valid from stop number 8010366 to stop number 8010097
- Infotext number (file INFOTEXT) = 0002905

5.3.8. *L-lines (optional):

The *L lines provide the line information for the service. If the *L-line is present in the data of a service, the appropriate line number for the route section entered in the *L-line is used. If the information on the route section is missing, the line number is used for the entire route.

File format:

Column	Type	Meaning
1−2	*CHAR*	*L
4−11	*CHAR*	line number
13−19	*[#]INT32*	(optional)Index or stop number from which the line number applies in the route.
21−27	*[#]INT32*	(optional) Index or stop number to which the line number applies in the route.
29−34	*[#]INT32*	(optional) Index for the xth occurrence or point of departure time
36−41	*[#]INT32*	(optional) Index for the xth occurrence or point of arrival time

Example:

```
*L 00000007 #6      8010097                                  %
```

Explanation:

- line type = *L
- line number = 7
- Valid from stop at position 7 in the route to stop 8010097

5.3.9. *R-line (optional

The *R-lines are used to provide a service with additional directional information.

File format:

Column	Type	Meaning
1−2	*CHAR*	*R
4−4	*CHAR*	(optional) code for direction (e.g. 0 = outward, 1 = return). This code is used for additional information such as line and direction related interchange times.
6−12	*CHAR*	(optinal)direction code (stop number or related number).
14−20	*[#]INT32*	(optional)Index or stop number from which the direction code applies in the route.
22−28	*[#]INT32*	(optional)Index or stop number to which the direction code applies in the route.
30−35	*[#]INT32*	(optional) Index for the xth occurrence or point of departure time
37−42	*[#]INT32*	(optional) Index for the xth occurrence or point of arrival time

If no other entries are made in the *R-line, the last stop in the route is used as the direction. If a direction code is entered, the software checks whether it can be interpreted as a stop number. In that case the stop name is used as a direction specification. Otherwise, the software looks for the appropriate entry in the file RICHTUNG.

Example:

```
*R          8010366 8010097                                  %
```

Explanation:

- Line type = *R
- The station's name is used for the indication of direction
- Valid from stop 8010366 to stop 8010097

5.3.10. **GR*-lines (optional):

The **GR* lines are used to provide some special stops of a service with additional route information. This kind of stops are called “border points”. Border points are useful to indicate e. g. the crossing of national borders, tariff boundaries or similar information. Border points are served by a service in general. A not serverd border point is called “virtual border point”. They are kept in the file GRENZHLT. A virtual border point has the purpose to indicate points which are important for generating connections and/or for generating prices. However it is not possible for a passenger to enter or to leave the service there. So it is not possible to select them as a start stop or a target stop by the passenger.

File format:

Column	Type	Meaning
1−3	*CHAR*	**GR*
5−11	*INT32*	specification of a (virtual) border point number
13−19	*[#]INT32*	(optional) Index or stop number of the last stop before the border point
21−27	*[#]INT32*	(optional) Index or stop number of the first stop after the border point
29−34	*[#]INT32*	(optional) Index for the xth occurrence or point of departure time
36−41	*[#]INT32*	(optional) Index for the xth occurrence or point of arrival time

If the service stops at the borderpoint in accordance with the timetable, the border stop number is also entered as the number of the first stop after the border point or last stop before the border point.

Example:

```
*GR 8705039 8700048 8800025                                    %
```

Explanation:

- Line type = **GR*
- Border stop at the stop 8705039
- The stop before the border stop is stop 8700048, the stop behind is stop (8800025)

5.3.11. *SH-Line (optional)

The *SH-lines suit the purpose, that some special stops of a route get other operating days. These seasonal stops can only be served at certain operating days.

File format:

Column	Type	Meaning
1−3	*CHAR*	*SH
5−10	*INT16*	(optional) Bitfield number, for the days, where the attribute is valid. Is this missing, the attribute is always valid.
12−18	*[#]INT32*	stop number or index number from the route (validity stop)
20−25	*[#]INT32*	(optional) Index for the xth occurence or departure time

Example:

```
*SH 8010097 023562                                        %
```

Explanation:

- Line Type = *SH
- Seasonal stop at the station 8010097
- The train only stops at the operating days, which are recorded in the key 023562

5.3.12. **CI*- and **CO*-line (optional)

The **CI*-line is used to define check-in times, the **CO*-line is used to define check-out times for a journey section. Check-in and check-out times influence the time a passenger needs for an interchange procedure. An interchange time can be a stop-related interchange time (see section 5.7), an interchange time between administrations (see section 6.7.2), or a line and direction related interchange time (see section 6.7.3). The check-out time of the train the passenger arrives with and the check-in time of the train the passenger wants to enter are added to this interchange time. Check-in and check-out times aren't used, if an interchange time between services (see section 6.7.5) exists for this two trains.

In oppositon to interchange times (UMSTEIGB, UMSTEIGV, UMSTEIGL, and UMSTEIGZ) check-in and check-out times are valid also at start and end of a journey or if the passenger is using a link between two stops.

File format:

Column	Type	Meaning
1−3	*CHAR*	**CI* or **CO*
5−8	*INT32*	Check-in/check-out time in minutes
10−16	*[#]INT32*	(optional) Index or stop number of the first stop with this check-in/check-out time
18−24	*[#]INT32*	(optional) Index or stop number of the last stop with this check-in/check-out time
26−31	*[#]INT32*	(optional) Index for the xth occurrence or point of departure time
33−38	*[#]INT32*	(optional) Index for the xth occurrence or point of arrival time

Example:

```
*CI 0005 8010366 8010097                                    %
```

Explanation:

- Line type = **CI*
- Check-in time = 5 minutes
- Valid from stop 8010366 to stop 8010097

5.3.13. Through coaches (**KW*-, **KWZ*- and **B*-lines):

A through coach is determined by those services which propel the coach on its line of route. Similar to the **Z*-lines a new through coach starts with a **KW*-line.

File format:

Column	Type	Meaning
1−3	*CHAR*	**KW*
5−9	*INT32*	ID number for the through coach

After this definition the services propelling the through coach are specified:

File format:

Column	Type	Meaning
1−4	*CHAR*	**KWZ*
6−10	*INT32*	ID number of propelling service
12−17	*CHAR*	Administration of propelling service
19−25	*INT32*	Stop number for the attachment
27−46	*CHAR*	(optional) stop name for the begin of the propelling
48−54	*INT32*	Stop number for the detachement
56−75	*CHAR*	(optional) stop name for the end of the propelling
77−82	*INT32*	(optional) depature time for the first stop
84−89	*INT32*	(optional) arrival time for the last stop

One can disable boarding and enlightening capabilites for certain parts of the schedule, for example to avoid disturbances in sleeperette coaches early in the morning.

Valid entries for the ”block type”are:
1: no boarding allowed
2: no alighting allowed
3: both boarding and alighting are blocked

File format:

Column	Type	Meaning
1−2	*CHAR*	**B*
4−4	*CHAR*	Sperrtyp (1/2/3)
6−12	*[#]INT32*	stop number from which the block is applicable.
14−20	*[#]INT32*	stop number up to which the block is applicable.
22−28	*[#]INT32*	stop number up to which the block is applicable.
30−35	*INT32*	time at former stop
37−42	*INT32*	time at latter stop

A through coach may have attribute information as well as different days of operation in comparison with the propelling services so additional *lines are necessary, i.e. **A VE* -lines.

Example:

```
*KW 00037                                                  %
*KWZ 00472 80____ 8500200 Zuerich HB          8000026 Basel Bad Bf         2215   2319
*KWZ 00470 80____ 8000026 Basel Bad Bf        8000152 Hannover Hbf         2340   0612
*KWZ 02746 80____ 8000152 Hannover Hbf        8000050 Bremen Hbf           0644   0754
*B 2      #0 8000026 002215 002319
*A VE 8500200 8000050 001339                               %
*A SL 8500200 8000050                                      %
```

Explanation:

The through coach is propelled by three services, it is detached at Basel Bad and Hannover. From the departure in Zuerich up to Basel, no alighting is allowed. The days of operation differ from those of the services. It’s a sleeperette coach.

5.3.14. Lines of the schedule:

All other lines are interpreted as lines of the schedule. They indicate the stops of the service.The stops must be arranged in chronological order.

File format:

Column	Type	Meaning
1−7	*INT32*	stop number
9−29	*CHAR*	(optional) stop name
30−35	*INT32*	arrival time at the stop (local time of stop)
37−42	*INT32*	departure time at the stop (local time of stop)
44−48	*INT32*	(optional) service number applicable from the stop
50−55	*CHAR*	(optional) administration applicable from the stop
57−57	*CHAR*	(optional) “X”, if this stop is given on the destination board of the service.

Remark:

“ ”	for no entry (only permissible for the arrival time of the first stop and the departure time of the last stop)
±5 digit figure	in the format PHHHMM. For a journey continuing beyond midnight, the times are counted further using 25 hrs, 26 hrs ... (a journey can continue for up to 984 hrs). The preceding sign “-” indicates informative times with boarding or alighting not allowed. The “+” sign can be replaced by a blank.

Attention:

If a station name occurs several times in a route (MGR service) the times must be different. The representation of a service is illustrated by an example with comments. The lines are consecutively numbered to aid comprehension.

Example:

```
  +-- This is the first column of a file
  _
1)*Z 00114 BVG_1B       011 020                          % 00114 BVG_1B
2)*G BUS 0053301 0053301                                 % 00114 BVG_1B
3)*A VE 0053301 0053352 000000                           % 00114 BVG_1B
```

```
 4)*A VE 0053252 0053301 000001                                  % 00114 BVG_1B
 5)*L 00001000                                                   % 00114 BVG_1B
 6)*R                                                            % 00114 BVG_1B
 7)0053301 S Wannsee DB                  02014                   % 00114 BVG_1B
 8)0053291 Wannseebrücke          02015  02015                   % 00114 BVG_1B
 9)0053202 Am Kl. Wannsee/Am Gr   02016  02016                   % 00114 BVG_1B
10)0053251 Seglerweg              02017  02017                   % 00114 BVG_1B
11)0053252 Koblanckstr.           02017  02017                   % 00114 BVG_1B
12)0053253 Colomierstr.           02018  02018                   % 00114 BVG_1B
13)0053254 Jugenderholungsheim    02018  02018                   % 00114 BVG_1B
14)0053255 Haus der Wannsee-Ko    02019  02019                   % 00114 BVG_1B
15)0053201 Wirtschaftshof         02020  02020                   % 00114 BVG_1B
16)0053255 Haus der Wannsee-Ko    02021  02021                   % 00114 BVG_1B
17)0053292 Krankenhaus Heckesho   02022  02022                   % 00114 BVG_1B
18)0053256 Zum Heckeshorn         02022  02022                   % 00114 BVG_1B
19)0053250 Straße zum Löwen       02023  02023                   % 00114 BVG_1B
20)0053251 Seglerweg              02023  02023                   % 00114 BVG_1B
21)0053260 Endestr.               02024  02024                   % 00114 BVG_1B
22)0053202 Am Kl. Wannsee/Am Gr   02025  02025                   % 00114 BVG_1B
23)0053291 Wannseebrücke          02026 -02026                   % 00114 BVG_1B
24)0053301 S Wannsee DB           02525                          % 00114 BVG_1B
25)*Z ...
```

Explanation:

1 The service starts at the position no. 1. It's the service 114 of the administration BVG_1B. This value pair identifies the service in the data set and is later used for references in the files VEREINIG, DURCHBI or UMSTEIGV. It is permissible for the pair (service number, administration) to appear several times in the data. In this case a reference in other files will lead to a multiple check, all possibilities are taken into account.

2 The service is a bus from stop 0053301 and back again. If the service category changes, a **G*-line is necessary for each section.

3 The **A VE*-lines indicate the operating days of the service. The operating day number "000000" means: service 114 runs daily on the section S Wannsee DB - Koblanckstrasse.

4 On the section Koblanckstrasse - S Wannsee DB the service runs according to the operating day index number 000001. The meaning of the operating day code is defined in the file BITFELD and could, for example, mean: Mo - Sa.

5 The service has the line number 1000. Since the stop numbers are missing and thus the range of applicability was not restricted, the line number applies to the entire line of route.

6 A **R*-line with no other entries: on the entire route, the name of the last stop is given as the direction.

7-22 The stops with arrival or departure times of the service. The name is optional but should always be added to help readability. The %-character in the 59th column is compulsory, everything after that is regarded as a comment (the service number is duplicated in the comment so that the entire service is found when searches are performed).

23 The minus sign indicates the depature time to be an internally operated time, not for the public, the boarding is not allowed here

24 The arrival time is on the next day, so the time indication 02525 will be converted to 00125.

25 A new service starts here...

5.3.15. Schedule data with regions (optional)

The schedule data may contain regions (see section 6.26):

File format:

Column	Type	Meaning
1−1	*CHAR*	“+”, marks a region line
3−8	*CHAR*	region number

Regions may be used to represent on demand services (teletaxis).

Each service must contain at least one stop. Every two region-lines must be seperated by at least one stop. The service may be ordered by customers to an arbitrary point within this region. The service passes the preceding stop, serves the ordered points within the region and continues its route with the subsequent stop.

To use this feature a Hafas server has to provide GIS services.

Example:

```
8500010 Basel SBB                 1500                  %
+123456 Region A                                        %
8503000 Zürich HB           1600                        %
```

Explanation:

The service passes the region 123456 between the stops 8500010 and 8503000. The service may be ordered to carry people to or from any arbitrary point within the region.

5.3.16. *TT-lines (optional)

*TT-lines mark services, which operate between a region and a set of stops. They may be thought of as feeder or distributor services for the given stops. The start or the end of a *TT-service is an arbitrary point within the region. Hafas ignores the sequence of the stops. So the customer who ordered the service will virtually be carried directly from his home to one of the stops or vice versa.

File format:

Column	Type	Meaning
1−2	*CHAR*	*TT
4−12	*INT32*	identifier

The service is used to travel from an address to a stop or vice versa.

Example:

```
*TT 123456789
```

Explanation:

The current service operates according to the *TT-rules. The *TT-identifier is 123456789.

Schedule data for *TT-services

Services having the *TT-flag use specific schedule lines which are similiar to the lines described in section 5.3.14. The service can be of type *Z or *T.

There are two different schedule line types for such services. The first defines a region (see section 6.26):

File format:

Column	Type	Meaning
1−1	*CHAR*	“+”, marks a region line
3−8	*CHAR*	region number
10−30	*CHAR*	(optional) region name

This line type may be used for the first and the last schedule line. If a line of this type is placed at the beginning of the schedule rows it is possible to use the service to travel from within the region to one of the stops. If the line is placed at the end of the schedule block, it is possible to travel from one of the stops to an arbitrary point within the region. If there is a **TT* -line there must be at least one line giving a region.

The second schedule line type defines the stops:

File format:

Column	Type	Meaning
1−7	*CHAR*	external stop number
8−13	*INT32*	arrival time at the stop (local time of stop)
15−20	*INT32*	departure time at the stop (local time of stop)

Example:

```
*Z 12345 987654                                    %
+123456 Region A                                   %
8013623 Uelzen                1400 1430            %
8001173 Lüneburg                   1500            %
+123456 Region A                                   %
```

Explanation:

The service may be used to travel between an arbitrary point within region 123456 and the stops 8013623 and 8001173. The arrival time at stop 8013623 is 2 pm. The departure time at stop 8001173 is 3 pm. It is permitted to travel to or from stop 8013623 and to travel from stop 8001173 but not to stop 8001173.

Example:

```
*T 12345 987654 0240 0450                                    %
+123456 Region A                                             %
8013623 Uelzen                1400 1430                      %
8001173 Lüneburg                   1500                      %
+123456 Region A                                             %
```

Explanation:

Here is an example for a *T service. The times at the stops give the earliest possible departure/arrival times. The service is available for four hours (240 minutes).

5.3.17. **E*-lines (optional)

**EN*-lines define a section in which interchanging is not allowed, which means it is not allowed to change within the defined section. **EI*-lines define a section in which it is possible to travel from any station to any other served station.

File format:

Column	Type	Meaning
1−3	*CHAR*	**E{I, N}* I interchanging allowed and N interchanging not allowed respectively.
5−11	*[#]INT32*	Index or stop number from which the section applies in the route.
13−19	*[#]INT32*	Index or stop number to which the section applies in the route.
21−26	*[#]INT32*	(optional) Index for the xth occurrence or point of departure time
28−33	*[#]INT32*	(optional) Index for the xth occurrence or point of arrival time

Example:

```
*Z 01504 80____    01                                    %
*G ICE 8000261 8002553                                   %
*EN 8011113 8010404                                      %
*A VE                                                    %
8000261 München Hbf                    01520             %
8000183 Ingolstadt Hbf          01557  01559             %
8000284 Nürnberg Hbf            01631  01636             %
8010309 Saalfeld(Saale)         01813  01815             %
8011956 Jena Paradies           01842  01844             %
8010240 Naumburg(Saale)Hbf      01906  01908             %
8010205 Leipzig Hbf             01946  01951             %
8011113 Berlin Südkreuz         02053  02055             %
8098160 Berlin Hbf (tief)       02105  02126             %
8010404 Berlin-Spandau          02133  02135             %
8002549 Hamburg Hbf             02305  02310             %
8002548 Hamburg Dammtor         02313  02315             %
8002553 Hamburg-Altona          02321                     %
```

Explanation:

Here is an example for a service with a section in which interchanging is not allowed, this section is valid from 8011113 Berlin Südkreuz to 8010404 Berlin-Spandau. So it is not allowed to travel from Berlin Südkreuz to Berlin-Spandau. The service can be used to travel to Berlin and from Berlin to Hamburg respectively.

Example:

```
*T 00001 VGS___ 0720 -0900                                  %
*EI 0000288 0000332                                         %
*G Ruf                                                      %
*A VE                                                       %
0000288 Böddenstedt, Ort (Sa         00800                  %
0000217 Böddenstedt, Abzweig  00800  00800                  %
0000209 Salzwedel, Böddenste  00800  00800                  %
0000364 Salzwedel, PVGS       00800  00800                  %
0000374 Salzwedel, Reitstadi  00800  00800                  %
0000230 Salzwedel, Sportplat  00800  00800                  %
0000617 Salzwedel, Dreilände  00800  00800                  %
0000360 Salzwedel, Lüneburge  00800  00800                  %
0000306 Salzwedel, Uelzener   00800  00800                  %
0000199 Salzwedel, Uelzener   00800  00800                  %
0000540 Salzwedel, Brunnenst  00800  00800                  %
0000332 Salzwedel, Kaufland   00800  00800                  %
```

Explanation:

The service can be used to travel from any served station to any other. So it is possible to travel from Salzwedel, PVGS to Salzwedel, Kaufland, but also to travel from Salzwedel, Kaufland to Salzwedel, PVGS. The negative clock density within the **T*-line defines the approximatet length of a journey in seconds.

5.4. Days of operation

To describe the situation precisely when a service operates and when not, two entries are required:

- the period of the timetable
- the operating days of the individual services (bit fields) related to this period

5.4.1. Basic data of the timetable period [ECKDATEN]

The period of the timetable is entered as follows:

- first day of the timetable,
- last day of the timetable,
- plain text containing the name of the timetable.

File format:

Line	Column	Type	Meaning
1	1−10	*CHAR*	start of timetable in the format DD.MM.YYYY
2	1−10	*CHAR*	end of timetable in the format DD.MM.YYYY
3	1*ff*	*CHAR*	timetable name

Example:

```
30.09.1996 start of timetable
01.06.1997 end of timetable
"Test timetable 1996/97"
```

5.4.2. Days of operation [BITFELD]

The operating days of a service can be represented as a bit field in which each bit corresponds to a day of the timetable period. A bit is 1 if the service runs on this day and 0 if it does not run. All bit fields which occur in the timetable period are assembled in the file BITFELD.

File format:

Column	Type	Meaning
$1-6$	*INT32*	bit field number (specified with leading zeros)
$8-103$	*CHAR*	bit field consisting of 96 hexadecimal digits. (ASCII format)

Remark:

- The number identifies a specific bit field. The numbering is 6-digit and starts at “000001”. Bit field “000000” is reserved for services which run daily.
- The bit field consists of 380 bits (days). The first bit represents the start of the timetable period. All unused bits after the last bit at the end of the timetable period are set permanently to “0”. To enable storage of the data in a more compact form, 4 bits are combined in one hexadecimal digit. In all, the bit field is therefore defined by 96 hexadecimal digits. A complete year’s timetable can therefore be represented.

For technical reasons 2 bits are inserted directly before the first day of the start of the timetable period and two bits directly after the last day at the end of the timetable period.

Example:

The timetable period starts on 27/09/1996 and ends on 22/05/1997. The operating day specification “runs every Saturday” is to be encoded. The 27/09/1996 is a Sunday.

```
So Mo Di Mi  Do Fr Sa So  Mo Di Mi Do  Fr Sa So Mo  Di Mi Do Fr  Sa So Mo Di  Mi Do Fr Sa
0  0  0  0   0  0  1  0   0  0  0  0   0  1  0  0   0  0  0  0   1  0  0  0   0  0  0  1
+--------+   +--------+   +--------+   +--------+   +--------+   +--------+   +--------+
```

```
   0          2          0          4          0          8         1
```

The bit field begins with 0204081... and the following data line results:

```
000001 02040810204081020408102040810204081 ...

   0        2        0        4        0        8        1        0        2        0
0 0 0 0 | 0 0 1 0 | 0 0 0 0 | 0 1 0 0 | 0 0 0 0 | 1 0 0 0 | 0 0 0 1 | 0 0 0 0 | 0 0 1 0 | 0 0 0 0
```

Now the bits before and after the timetable period must be set, giving:

```
000001 C08102040810204081020408102040810204081020 ...

   C        0        8        1        0        2        0        4        0        8
1 1 0 0 | 0 0 0 0 | 1 0 0 0 | 0 0 0 1 | 0 0 0 0 | 0 0 1 0 | 0 0 0 0 | 0 1 0 0 | 0 0 0 0 | 1 0 0 0
```

5.5. Means of transport or service categories [ZUGART]

The **G*-line in the timetable file is used to define the means of transport or service categories (Bus, U-Bahn, ICE, IC, EC, etc.).

The file contains the following for each means of transport (or each service category):

- the 3-digit code as used in the timetable files,
- definition of the quality of the means of transport or service. A number between 0 and 13 is used for this purpose. A “0” is the highest quality, a “13” the lowest quality. By defining the quality of a means of transport it is allocated to classes. In *Hafas* the search can be influenced by selecting or deselecting these classes,
- definition of a tariff group A, B, C etc. In case one definition is set in one line all other lines in the file must obtain a definition,
- definition of the representation for output. A “0” stands for the output of both service category and service or line number, “1” for the output of service category only, “2” for output of service or train number only and “3” for no output. In case the service category should be replaced by the operator add another 4 to obtain the related display mentioned before, i.e. 4 for operator and number, 5 for operator only and so on,
- the service category for the output. The service category text must not exceed 8 characters,
- number indicating whether the service is free of supplements or not,
- code whether local service, ship or airplane,
- index to logo name for this means of transport
- full name for train type or service category, not limited in length, or index to internationalized name

Comments exceeding to the right are not permitted for they are considered to be part of the train type name.

File format:

Column	Type	Meaning
1−3	*CHAR*	service category code in the data
5−6	*INT16*	class (from 0 to 13). Required to limit a search to certain classes (e.g. without ICE)
8−8	*CHAR*	tariff group A-H
10−10	*INT16*	control of output: **0** service category and number **1** service category only **2** number only **3** no output **+4** administration in place of the service category
12−19	*CHAR*	service category for output. If no output is required, enter “-”.
21−21	*INT16*	Supplement: **0** no supplement **1** supplement due, depends on context **2** supplement generally due
23−23	*CHAR*	Flags: **“N”:** service category belongs to local transport. **“B”:** means of transport is a ship **“F”:** means of transport is an airplane
25−28	*[$]INT32*	(optional) index number of logo name for train type, limited to values between 0 and 999
30−33	*[#]INT32*	(optional) index number of internationalized full description of train type, limited to values between 0 and 999. Instead of index number a text is possible as well. This text will be used for all languages.

Remark:

The service category “UUU” for unknown service category is mandatory. Variations of the assignment of a train category to tariff group resp. train category to product group are possible, i.e. the assignment of a train category to a product class and to a tariff group is independent to eachother. A definition of a tariff group results exclusively based on the train category. Other arrangements are not supported.

Example:

```
N    3 A 0 N        0 N Nahverkehrszug
```

Explanation:

- Train type = N
- Class = 03
- Tariff group = A
- The services of this type are displayed with train type and number
- The displayed train type is N
- The services of this type need no supplement
- They are local services
- The full description is “Nahverkehrszug”

5.5.1. Additional information in file ZUGART

In the bottom part it is possible to define internationalized texts which are visible in the *Hafas*-GUI. Furthermore here is the information located set by the index numbers for logo names and full descriptions.

Product texts can be attributed to the product classes. The product texts appear in the timetable information and indicate the according product class there. Classes denoting train types/categories may obtain additional text to be displayed in *Hafas* which can be defined separately for each language. The usage of product texts is optional. The same applies to options and associated texts, to tariff groups and full descriptions of means of transport.

The beginning of the text definitions is marked with the tag `<text>`. All lines following are regarded as class text to be displayed. Each data set starts with a line defining the language through a tag. The following language tags are possible at the moment:

- `<German>`
- `<English>`
- `<French>`
- `<Spanish>`
- `<Italian>`
- `<Japanese>`
- `<Dutch>`
- `<Polish>`
- `<Norwegian>`
- `<Swedish>`
- `<Turkish>`
- `<Hungarian>`
- `<Danish>`
- `<Irish>`

The language tag is followed by the text lines concerning each class, starting with “class00” up to “class13”.

File format:

Column	Type	Meaning
$1-7$	*classXX*	Product class XX (XX between 00 and 13)
$9ff$	*CHAR*	Class text

To enable calculation on specific characteristics variables starting with “option00” up to “option04” are possible. The result depends on the version of *Hafas* .

File format:

Column	Type	Meaning
$1-8$	*optionXX*	Option definition XX (XX between 00 and 04)
$10ff$	*CHAR*	Option text

For tariff groups the declarations “tariff00” up to “tariff07” are possible. The associated texts correspond to the tariff groups A to H in the upper part of this file. The usage of tariff groups during the connection search depends on the version of *Hafas* .

File format:

Column	Type	Meaning
$1-8$	*tariffXX*	Tariff group XX (XX between 00 and 07, corresponds to the tariff groups A to H)
$10ff$	*CHAR*	Tariff group text

If a product class, an option or a tariff group shall not be displayed in the dialog then the associated text has to be “-”. For each option it is possible to force “direct connections only” when activated. This may be denoted by an additional * (not being displayed) after the text.

The full description of the means of transports terminates the language dependent definitions.

File format:

Column	Type	Meaning
$1-11$	*categoryXXX*	Train type text number XXX (XX between 000 and 999)
$13ff$	*CHAR*	Train type text

Subsequent to the language dependent definitions follows the declaration of the (language independent) train type logo names. This block starts with `<picture>`.

File format:

Column	Type	Meaning
$1-10$	*pictureXXX*	Train type logo number XXX (XX between 000 and 999)
$12ff$	*CHAR*	Logo name associated to the train type

The logo numbers 000 to 013 are used as default for the product classes 00 to 13. If there is no logo number defined for a certain train type, the default logo for the associated product class is used.

Example:

```
<text>
<German>
class00             "ICE-Züge"
class01             "Intercity- und Eurocityzüge"
class02             "Interregio- und Schnellzüge"
class03             "Nahverkehr, sonstige Züge"
class04             "S-Bahnen"
class05             "Busse"
class06             "Schiffe"
class07             "U-Bahn"
class08             "Straßenbahn"
class09             "Anrufpflichtige Verkehre"
option00            "nur Direktverbindungen"
option01            "nur Züge mit Schlafwagen*"
option02            "nur Züge mit Liegewagen*"
option03            "nur Züge mit Fahrradbeförderung"
option04            "-"
category001         "Regionalzug"
category002         "Niederflurwagen"
...

<English>
class00             "ICE-Trains"
class01             "Intercity- and Eurocitytrains"
class02             "Interregio- and Fast trains"
class03             "Regional and other trains"
class04             "S-Bahn"
class05             "Busses"
class06             "Boats"
class07             "Underground"
class08             "Tram"
class09             "Services requiring tel. registration for passengers"
option00            "only direct connections"
option01            "only trains with sleepers*"
option02            "only trains with couchettes*"
option03            "only trains with bicycle transport"
option04            "-"
category001         "Regional train"
category002         "low-loader wagon"
...

<French>
class00             "Trains ICE"
...

<picture>
picture001          "ice.gif"
picture002          "ic.gif"
...
```

5.6. Interchanges between stops [METABHF]

In many cases it is not enough to provide interchange time information within a stop as links between different stops (e.g. in Paris between the stations Paris-Nord, Paris-Est) exist as well. This means that between the stops, connection services exist which are covered by means of transport not contained in the database (e.g. taxi, Metro, on foot, etc.). In addition, it is often useful to consider not one but several stops as the starting point (e.g. a central bus station comprises several stop positions which are not served by all lines). Therefore two information blocks are defined in this file:

Links:

For individual pairs of stops, special links can be defined (e.g. for Paris-Nord and Paris-Est) if a link exists in between (e.g. by footpath, Underground, etc.). For this purpose, the time required is also specified. The following are required for the representation:

- the stop numbers to generate the link
- the time required for the link
- explanatory texts for the link (optional)
- operating days for the link (optional)
- operating times for the link (optional)
- a value (virtual interchange) for the link (optional)
- a class for the link (optional)
- flags indicating that the link shall not be displayed at start and destination of a journey or during a journey (optional)
- a flag which marks the link as part of a guaranteed interchange (optional)
- the exact walking distance (optional)

Some of these optional features cannot be used for walking links which connect different pools.

Stop groups:

The second information block consists of entries which assemble stops into groups. There are different types of “groupings“‘, each having its specific meaning. The classical meaning of an entry reads as “In case stop X is entered as start/destination, search from/to Y and Z as well”, i.e. X is substituted by Y or Z. Two further types of entries are derived from this type: The first type means “In addition, show a walking link from X to Y or Z”, the second type “Substituting X by Y (or Z) will cost an extra (“virtual“) interchange”. Another type of grouping reads as “List all services operating to/from stop Y or Z in the station board of stop X”.

Each entry appears as follows:

- number of the stop group
- a list of numbers of alternative (equivalent) stops. Each stop within the list of equivalences may be preceded by a flag which indicates the type of this entry: a classical substitution, a substitution combined with a walking link, a substitution combined with an extra interchange, or grouping for station boards.

5.6.1. Links

The “link” between two stops is represented by entering the two stop numbers and a time in minutes; the link is valid only in the direction from stop 1 to stop 2. If the opposite direction is also available, this link must be listed separately.

File format:

Column	Type	Meaning
1−7	*INT32*	Stop number 1
9−15	*INT32*	Stop number 2
17−19	*INT16*	Duration of link in minutes
20−20	*CHAR*	“S” as separator between minutes and seconds (optional)
21−22	*INT16*	Duration of link in seconds (optional)

This entry defines a link from stop 1 to stop 2. The flag “F”, which was formerly required to make this walking link shown in case stop 1 is the start or stop 2 is the destination of the connection request, may still be used (see section 5.6.12 for implications of the flag “F“). However, we recommend to use F-equivalences instead (see section 5.6.12).

For the time being the older format with an attribute between stop 2 and flag “F“ is supported, too.

Example:

```
% Abgeordnetenhaus von Berlin -> S+U Potsdamer Platz
0012105 0100020 005
0100020 0012105 004S30
```

Explanation:

It is possible to cover the route section from the stop Abgeordnetenhaus to the S-Bahn or U-Bahn station at Potsdamer Platz by foot in approximately five minutes. The second line describes the walking link in opposite direction, which requires only four minutes and thirty seconds.

For each link several *-lines can be defined:

5.6.2. *A-lines (optional)

Several attributes for a link can be given. The software looks for the associated text in the file ATTRIBUT.

File format:

Column	Type	Meaning
1−2	*CHAR*	*A
4−5	*CHAR*	Attribute code

Example:

```
0012105 0100020 005
*A B1
*A B2
```

Explanation:

This link has the attributes B1 and B2 from the file ATTRIBUT.

5.6.3. *V-lines (optional)

If a link can't be used for all days of a timetable period, it is possible to restrain its operating days. For each link only one operating period can be defined. But it is possible to enter the same link several times with different operating days.

File format:

Column	Type	Meaning
1−2	*CHAR*	*V
4−9	*INT32*	Operating day number for the days on which the service takes place.

Operating days cannot be defined for walking links which connect different pools.

Example:

```
0012105 0100020 005
*V 000012
```

Explanation:

The link has the operating day number 000012 (see the file BITFELD).

5.6.4. *O-lines (optional)

Some links have opening hours (lifts, moving stairs, . . .). These opening hours can be defined as follows: The start time has to be between 00:00 and 24:00. The last time has to be in the interval of 24 hours after the start time.

For each link only one interval of opening hours can be defined. But it is possible to enter the same link several times with different opening hours. A combination of operating days and opening hours is possible.

File format:

Column	Type	Meaning
1−2	*CHAR*	*O
4−7	*INT16*	start of the opening hours
9−12	*INT16*	end of the opening hours

Opening hours cannot be defined for walking links which connect different pools.

Example:

```
0012105 0100020 005
*O 1625 2813
```

Explanation:

The link is open between 16:25 and 04:13 Uhr (following day).

5.6.5. **U*-lines (optional)

If one link is espacially unfavourable, it is possible to provide a value to it. During the search for a connection this value is used in the following way: If the link has the value two and it is possible to replace it by two trains, this new connection would be evaluated as good as the connection with the link instead of the two trains.

Only one value is allowed for each link.

File format:

Column	Type	Meaning
1−2	*CHAR*	**U*
4−4	*INT16*	Value for the link (0-7)

Example:

```
0012105 0100020 005
*U 2
```

Explanation:

The link is evaluated as two trains.

5.6.6. *C-lines (optional)

If some links has to be used consecutively, it is maybe undesirable to evaluate the number of transfers between the links during the search for a connection. For example the way from an underground to a train platform consists of a footpath, moving stairs, a footpath, and a lift. To avoid this it is possible to use the same class for such footpaths.

Only one class is allowed for each link.

File format:

Column	Type	Meaning
1−2	*CHAR*	*C
4−8	*INT32*	Class for the link (1-65000)

Example:

```
0012105 0100020 005
*C 30000
```

Explanation:

The link has the class 30000.

5.6.7. *I-lines (optional):

On the basis of the information text lines it is possible to supplement a footpath with additional information in plain text. Whilst a particular text is permanently assigned to an attribute code (see *A) it is possible to indicate various texts in relation to an infotext code. Several information text lines for a link can be given.

File format:

Column	Type	Meaning
1−2	*CHAR*	*I
4−5	*CHAR*	Infotext code. If "'XI"' is used, the infotext is treated as if containing XML.
7−13	*INT32*	(optional) Number of the information text from the file INFOTEXT.

Information texts cannot be defined for walking links which connect different pools.

Example:

```
0012105 0100020 005
*I FW 0002905
```

Explanation:

The link has an information text with the code FW and the number 2905 from the file INFOTEXT.

5.6.8. *B-lines (optional):

Using this flags it is possible to define that the link shall not be presented to the customer. These links will be used searching a connection but they will be hidden in the output.

It is possible to differentiate between the use of a link at start or destination of a journey and the use during a journing as a transfer between to services.

It is also possible to define links which are not used while searching a connection (hidden links). This links can be shown as additional information if a connection starts or ends at a V-equivalent station (see section 5.6.12).

File format:

Column	Type	Meaning
1−2	*CHAR*	*B
4−4	*INT16*	possible values: **1** at start and destination only **2** during a journey only **3** hide always **4** ignore link while searching a connection

Links which connect different pools cannot be ignored while searching a connection.

Example:

```
0012105 0100020 005
*B 3
```

Explanation:

The link has a duration of 5 Minutes but it never will be displayed.

5.6.9. **G*-lines (optional):

The **G*-line marks a walking link as a “guaranteed walking link“ between two services or lines, respectively. This means that the walking link connects two stops, the first being the place of arrival of a feeder, the second being the place of departure of a connecting service. But this link can only be used if it is combined with specific services (see section 6.7.6) or lines (see section 6.7.4), and it must not be used in any other case. In order to specify the allowed combinations within the UMSTFWL or UMSTFWZ file, each guaranteed walking link must be provided with a number which is unique among all guaranteed walking links.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	**G*
$4-10$	*CHAR*	Number of guaranteed walking link

Example:

```
0012105 0100020 005
*G 0000001
```

Links which connect different pools cannot be guaranteed.

Explanation:

The walking link from stop 0012105 to 0100020 is a guaranteed walking link. The number 0000001 serves as a reference to this walking link.

5.6.10. **L*-lines (optional):

As the real walking distance between two stops may considerably differ from the air-line distance in between, the walking distance may be specified by a **L*-line.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	**L*
$4-10$	*CHAR*	Walking distance in metres

Example:

```
0012105 0100020 005
*L 0000250
```

Explanation:

The walking distance from stop 0012105 to 0100020 amounts to 250 metres.

5.6.11. **E*-lines (optional):

A **E*-line assigns a number to the footpath. These numbers can be used in HAFAS to reference the footpath.

File format:

Column	Type	Meaning
1−2	*CHAR*	**E*
4−12	*INT32*	Footpath number

One footpath may have several numbers. One number can be assigned to several footpaths, i.e. one number can reference more than one footpath.

Example:

```
0012105 0100020 005
*E 123456789
```

Explanation:

The number 123456789 is assigned to the footpath between the stops 0012105 and 0100020.

5.6.12. Stop groups

Among all lines of format 1 the lines which describe the “grouping“ of stops can be listed. The task is

- to permit alternative stops as start, destination, or via instead of or in addition to the given stop
- to show the services operating to/from the alternative stops in the station board of the given stop.

The alternative stops are “equivalent“ or “synonymous“ to the given stop, which serves as a group term of its alternative stops.

The interval of an equivalence depends on its type. Four different types may be defined:

Type S: Start or destination equivalence. Serving as start, destination, or via of a journey, the group term is substituted by its S-equivalent stops, which are equal in terms.

Type V: Start or destination equivalence including “virtual“ interchanges. Similar to type S, but substituting will increase the number of interchanges.

Type F: Uses walking links (footpaths). The F-equivalent stops will be reached by a walking link from the stops which are S- and V-equivalent to the group term, if the latter is the start or destination of a connection request.

Type B: Refers to station boards. The station board of the group term contains exactly those services which operate to/from one of the B-equivalent stops. If there is no B-equivalence to a certain stop, then its F-, S-, and V-equivalent stops will be used as B-equivalent stops.

Type H: Main mast. Masts are different parts of the same stop (e.g. platforms of a station). Two stops are masts if they are S-equivalent and have the same name. When two masts should have different names or shouldn’t be S-equivalent, one can define a main mast for each of them. The stops are identified as masts if they have the same main mast. Maximal one main mast can be defined for each stop. If some features (infotexts, attributes, . . .) shall be valid for all masts of a station, it is enough to define them for the main mast only.

Remark:

Former definitions of stop groups considered only one type of equivalences, namely type S.

File format:

Column	Type	Meaning
$1-7$	*INT32*	The number of the group term
$8-8$	*CHAR*	Fixed symbol ":"
$10-10$	*CHAR*	The type of the 1st equivalence: B, F, V, or blank
$11-17$	*INT32*	The number of the 1st equivalence
$19-19$	*CHAR*	The type of the 2ndequivalence: B, F, V, or blank
$20-26$	*INT32*	The number of the 2nd equivalence
$28ff$		Type and number of the next equivalence ... etc.

- Leaving the type of the equivalence blank characterizes S-equivalences.
- Each group term must appear in the stop file (BAHNHOF) but may be a virtual stop (i.e. it is not essential that services are operated to or from it).
- Each stop within the stop file (BAHNHOF) which does not appear in METABHF file is S-equivalent to itself and has no further equivalences.
- For each group term in METABHF there must be a S- or a V-equivalence. It is not possible to define solely B-, F-, or H-equivalences for a group term.
- If a group term itself is required as a start, destination, or via, it must be defined to be S-equivalent to itself. Similar, if the group term's station board is expected to show all services operating to/from it, then the group term has to be B-equivalent to itself. On the other hand, no group term may be F- oder V-equivalent to itself.
- METABHF may contain two or more lines corresponding to the same group term. These lines will be regarded as a single one whose equivalence list is just the union of the group term's equivalence lists.
- In order to define stop Y to be F-equivalent to stop X, a walking link from X to Y is must be defined in METABHF.
- If the link from stop X to stop Y carries a flag "F" and Y is S- oder V-equivalent to X, then Y is considered to be F-equivalent to X.

Example:

```
% S+U Potsdamer Platz: S+U Potsdamer Platz, Abgeordnetenhaus
0012105: 0012105 0100020
% Frankfurt(Main)Hbf: Frankfurt(Main)Hbf, Frankfurt(Main)Süd
8000105: 800105 V8002041
% Frankfurt(Main)Hbf: Frankfurt Hbf (tief)
8000105: F8098105
% Frankfurt Hbf (tief): Frankfurt(Main)Hbf
8000105: H8000105
8098105: H8000105
```

Explanation:

If “S+U Potsdamer Platz“ is specified as the start stop, the search starts from the stops “S+U Potsdamer Platz“ and “Abgeordnetenhaus“ on an equal basis.

If the start stop is “Frankfurt(Main)Hbf“, the connection search will use “Frankfurt(Main) Süd“ as additional start stop, but if a connections starts at “Frankfurt(Main) Süd“, then its number of changes will be increased by a fixed value (a “virtual interchange“), which is important with respect to comparison of “competing“ connections.

The start stop being “Frankfurt(Main)Hbf“, the walking link from its S- resp. V-equivalences “Frankfurt(Main)Hbf“ resp. “Frankfurt(Main) Süd“ to “Frankfurt Hbf (tief)“ may be used as the first section of a connection.

If METABHF contains no further lines referring to the group term 8000105 “Frankfurt(Main) Hbf“, then the station board of “Frankfurt(Main) Hbf“ will show exactly those services which operate to/from “Frankfurt(Main) Hbf“, “Frankfurt(Hbf) tief“, or “Frankfurt(Main) Süd“.

”‘Frankfurt(Main)Hbf”’ is the main mast for ”‘Frankfurt(Main)Hbf”’ and ”‘Frankfurt Hbf (tief)”’. The two stops are masts although they have different names.

5.7. Stop-related interchange times [UMSTEIGB]

All interchange procedures, e.g. between services, require time. Therefore, a global standard interchange time must be defined which specifies how much time the traveller needs for the interchange.

For the standard interchange time a difference is made for changes between classes 0 up to 1 and interchanges between other classes. Individual interchange times between transport lines in complex interchange systems are mapped in the file UMSTEIGL by splitting into individual stops and links via footpaths (“Metakanten”) or by the definition of line-related interchange times.

Apart from the standard interchange time, a special interchange time can be defined for individual stops to take into account the physical layout of the stop.

The first line of the file contains the standard interchange times for all interchange points for which no special interchange times have been set in the following. The first line must be present, all other lines are optional.

File format:

Column	Type	Meaning
$1-7$	*INT32*	the number of the stop
$9-10$	*INT16*	interchange time for classes 0 to 2 (max. 60 minutes)
$12-13$	*INT16*	interchange time between all other classes
$15ff$	*CHAR*	(optional) plain text of stop name

The virtual stop number 9999999 is entered in the first line of the file to define the default value.

Example:

```
9999999 2 3 STANDARD
5800010 5 5 Basel SBB
```

Explanation:

At all stops not mentioned in the following, the minimum interchange time for an IC-IC (class 2) interchange procedure is 2 minutes. For all other interchange processes the minimum interchange time is 3 minutes. In Basle SBB the minimum interchange time for all interchange processes is 5 minutes.

6. Optional data files

The files described as follows are not compulsory for the operation of *Hafas* . Depending on the characteristics of the transport network, they are, however, necessary to obtain high quality of information. Moreover, additional information is made available to the user.

6.1. Additional data for stops [BHFART]

This setting enables to define restrictions in the selection list for stops within *Hafas* or denotes additional text for the stops.

- Code B defines a restriction for a stop, that must not serve as origin, destination or via for a journey,
- Code A defines information text by attribute,
- Code I defines text from a information text,
- Code H defines a “hailing section”.
- Code T defines the types of individual traffic (footpath, bike, taxi, ...) which can be used for the routing between any location and the stop.

The file is optional. In case the file is missing all stops may be used in the selection list and no additional text is available.

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number.
$9-9$	*CHAR*	B
$11-12$	*INT16*	Restriction in selection list. Values between 0 and 7.
$14-15$	*INT16*	Restriction for individual routing. Values between 0 and 7.
$17ff$	*CHAR*	Stop name (optional)

Example:

```
8504419 B  3  3 Biel Mett
8504300 B  4  0 Biel/Bienne
8000122 B  0  0 Bietingen
```

Explanation:

Biel Mett is used in a selection list as a via stop only (first 3), additionally Biel Mett may not serve as a stop for routing from an adress at start or destination of a journey (second 3). **Biel/Bienne** is used as origin/destination only and it may be routet to. **Bietingen** has no restrictions (0 may be omitted for it's the default value). If a stop shouldn't be available in a certain choice, it's possible to define exactly in this file, which choice is disabled. This stoppage is done bit by bit, at present three bits are used for that:

- bit 0 (value 1): stop excluded from selection list for origins,
- bit 1 (value 2): stop excluded from selection list for destinations,
- bit 2 (value 4): stop excluded from selection list for vias.

A combination of these definitions can be achived by addition of the values, i.e. value 3 denotes a via stop only.

The same bits are used to code restrictions for routing at start or destination of a journey.

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number
$9-9$	*CHAR*	A
$11-12$	*CHAR*	Attribute code
$14-19$	*INT32*	Bitfield number (optional)

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number
$9-9$	*CHAR*	I
$11-12$	*CHAR*	Infotext code. If "'XI"' is used, the infotext is treated as if containing XML.
$14-20$	*INT32*	Infotext number
$22-27$	*INT32*	Bitfield number (optional)

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number
$9-9$	*CHAR*	H
$11-15$	*INT16*	Stop - Flag. If the value 16 is filled in, it is a “hailing section”.
$17-17$	*CHAR*	Stop Name

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number or "'@@@@@@@" for all stops.
$9-9$	*CHAR*	T
$11ff$	*CHAR*	A set of characters from A to H which define the allowed types of individual traffic for the routing between any location and the stop. The first character can be a “!”. In this case the characters from A to H define the forbidden types of individual traffic.

The character from A to H represent the following types of individual traffic:

- A: footpath
- B: bike
- C: car
- D: park-and-ride
- E: taxi
- F: taxi at a taxi stand
- G: unused
- H: unused

For all stops all types of individual traffic are allowed by default.

Example:

```
@@@@@@@ T !D
8504419 B !CD
8504300 T ABCDEFGH
8000122 T B
```

Explanation:

For the routing between any location and any stop all types of individual traffic are allowed, except the routing for the park-and-ride-search. Certain rules are defined for the following stops:

- For the stop 8504419 all types of individual traffic are allowed, except the routing for the park-and-ride-search and the routing by car.
- For the stop 8504300 all types of individual traffic are allowed.
- For the stop 8000122 the routing by bike is allowed only.

Examples for definitions are given in the appendix.

6.2. Stop attributes and meta attributes [BHFATTR]

This file lists attibutes for stops to be displayed in *Hafas* . In combination with the files BHFART and INFOTEXT the texts for attributes can be linked with various information texts.

6.2.1. Stop attributes

Line type 1:

File format:

Column	Type	Meaning
$1-2$	*CHAR*	Attribute code
$4-6$	*INT16*	Priority for display in *Hafas* . The lower the figure is the higher is the priority. Special operation for values equal/lower than 2. Values between 0 and 999.
$8-10$	*INT16*	Fine sort of attributes of the same priority. The lower the figure is the higher is the priority. Values between 0 and 99.
$12ff$	*CHAR*	description text (max. 70 characters)

Examples are in the appendix

- porter service telephone number,
- opening hours (multilingual).

6.2.2. Stop meta attributes

Stop attributes can get assigned various meta attributes. The codes of the meta attributes consist of the character ”+”and any two printable characters, with the exception of (one or more) spaces and codes ending in two capital characters. The latter are reserved for internal use in *Hafas*.

Line type 2:

Describes the texts of meta attributes as well as their priorities.

File format:

Column	Type	Meaning
$1-3$	*CHAR*	Meta attribute code
$5-7$	*INT16*	Priority for display in *Hafas* . The lower the figure is the higher is the priority. Special operation for values equal/lower than 2. Values between 0 and 999.
$9-10$	*INT16*	Fine sort of attributes of the same priority. The lower the figure is the higher is the priority. Values between 0 and 99.
$12ff$	*CHAR*	description text (max. 70 characters)

Example:

```
+SA 740 11 Search attribute
```

Explanation:

The meta attribute ”‘+SA”’ is created for marking attributes as ”relevant for searching”.

Line type 3:

Assigns one or more meta attributes to an attribute.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	Attribute code
$3-3$	*CHAR*	Fixed character ":"
$5ff$	*CHAR*	List of meta attribute codes, separated by spaces

Example:

```
HB: +bg +HA
HB: +wr
HR: +bg +wr
```

Explanation:

The meta attributes with the codes "'+bg"', "'+HA"' and "'+wr"' get assigned to the attribute with the code "'+HB"'. The attribute "'HR"' gets assigned meta attributes "'+bg"' and "'+wr"'.

Line type 4:

Assigns one or more meta attributes to another meta attribute.

File format:

Column	Type	Meaning
$1-3$	*CHAR*	Attribute code
$4-4$	*CHAR*	Fixed character ":"
$6ff$	*CHAR*	List of meta attribute codes, separated by spaces

Example:

```
+nb: +bf
```

Explanation:

The meta attribute with the code "'+bf"' gets assigned to the meta attribute with the code "'+nb"'.

6.3. Additional attributes and meta attributes[ATTRIBUT]

The file with attributes is necessary if you have used **A*-lines in the timetable data. The explanation text is given and whether the attribute is valid for part or complete sections. In combination with the files ATTRIBUT and INFOTEXT the texts for attributes can be linked with various information texts.

6.3.1. Additional attributes

The following definitions are used:

- the two-digit attribute code,
- to which stops the attribute applies when the attribute text is displayed,
- the plain text of the attribute,
- the priority for output in a list,
- fine sorting of the attribute output.

The **A*-lines in the timetable data contain the attribute code only. It's essential to provided the appropriate plain text for output. Moreover, it is possible to define whether an attribute is to apply to a journey section or to the departure or arrival station of the section.

Line format type 1:

Defines the stop affiliation and the attributes.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	Attribute code
$4-4$	*INT16*	stop affiliation ("1" for departure time, "2" for arrival time, 3 for both times of a stop,"0" for the section to the next stop)
$6-8$	*INT16*	attribute output priority in *Hafas* , low values have greater priority than large. In case several attributes must be displayed higher priority attributes are output first.
$10-11$	*INT16*	fine sorting of the attribute output, low values have greater priority. In case attributes of the same priority appear, the output depends on the significance of the fine sorting.
$13ff$	*CHAR*	attribute text (max. 70 characters) ending with "#"

Example:

```
B1 1 100 10 Stop on request only #
K1 0 200 10 1st class only #
LW 0 300 10 Couchette#
TL 0 300 11 Route section couchette#
```

Explanation:

“B1” stands for “stop on request ” and due to the “1” relates to the respective departure time of a stop.

“K1” stands for “1st class only” and relates to the journey section.

“LW” stands for “couchette” and “TL” for “Route section couchette” (i.e. the couchette is not available for the complete route section).

Line format type 2:

Defines how the output has to take place.

After the plain text has been specified for all attribute codes, there are further entries in this file which define the choice of output texts. The purpose of these mappings is greater flexibility in assigning abbreviations and defining texts if the attribute information applies only to part of sections of the route.

A mapping is a triplet consisting of:

- the attribute code which appears in the schedule,
- the attribute code which is to be output if the input attribute does not apply to the entire section of the service,
- the attribute code which is to be output if the input attribute applies to the entire section of the service.

In general, “--” stands for “no output”. To separate lines of type 2 from lines of type 1, a “#” is placed in front of lines of the type 2.

File format:

Column	Type	Meaning
1−1	*CHAR*	Fixed character “#”
3−4	*CHAR*	Attribute code in the timetable data
6−7	*CHAR*	“--”, if this attribute is to be suppressed in the output, otherwise attribute code for output of the part route section
9−10	*CHAR*	“--”, if this attribute is to be suppressed in the output, otherwise attribute code for output of the full route section. If the attribute code is only single-digit, this does not have to be filled with blanks.

Example:

```
# LW TL LW
```

Explanation:

In case the attribute “LW” appears in the timetable data, the plain text of the attribute “TL” is displayed if it only applies to the part route section. On the full route section, the text for “LW” is displayed.

6.3.2. Meta attributes

Attributes can get assigned various meta attributes. The codes of the meta attributes consist of the character ”+”and any two printable characters, with the exception of (one or more) spaces and codes ending in two capital characters. The latter are reserved for internal use in *Hafas*.

Line type 3:

Describes the texts of meta attributes as well as their priorities.

File format:

Column	Type	Meaning
$1-3$	*CHAR*	Meta attribute code
$5-7$	*INT16*	Priority for display in *Hafas* . The lower the figure is the higher is the priority. Special operation for values equal/lower than 2. Values between 0 and 999.
$9-10$	*INT16*	Fine sort of attributes of the same priority. The lower the figure is the higher is the priority. Values between 0 and 99.
$12ff$	*CHAR*	description text (max. 70 characters)

Example:

```
+SA 740 11 Search attribute
```

Explanation:

The meta attribute ”‘+SA”’ is created for marking attributes as ”relevant for searching”.

Line type 4:

Assigns one more more meta attributes to an attribute.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	Attribute code
$3-3$	*CHAR*	Fixed character ":"
$5ff$	*CHAR*	List of meta attribute codes, separated by spaces

Example:

```
HB: +bg +HA
HB: +wr
HR: +bg +wr
```

Explanation:

The meta attributes with the codes "'+bg"', "'+HA"' and "'+wr"' get assigned to the attribute with the code "'+HB"'. The attribute "'HR"' gets assigned meta attributes "'+bg"' and "'+wr"'.

Line type 5:

Assigns one more more meta attributes to another meta attribute.

File format:

Column	Type	Meaning
$1-3$	*CHAR*	Attribute code
$4-4$	*CHAR*	Fixed character ":"
$6ff$	*CHAR*	List of meta attribute codes, separated by spaces

Example:

```
+nb: +bf
```

Explanation:

The meta attribute with the code "'+bf"' gets assigned to the meta attribute with the code "'+nb"'.

6.4. Stop interchange priorities [BFPRIOS]

If there are several alternative interchange possibilities along a route because two lines run parallel, for example, the choice of interchange point can be influenced by defining an interchange priority. The interchange priority is a value between 0 and 16, in which 0 means highest priority and 16 lowest priority. As default, the interchange priority is set by the program to 8.

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number
$9-10$	*INT16*	interchange priority of the stop. Only interchange priorities between 0 and 16 are accepted.
$12ff$	*CHAR*	stop name (optional).

Example:

```
8504419    8    Biel Mett
8504300    4    Biel/Bienne
8000122    8    Bietingen
```

Explanation:

If an interchange is possible in Biel Mett, Biel/Bienne or Bietingen with services otherwise equal, Biel/Bienne is given preference.

6.5. Train information texts [INFOTEXT]

This file contains the train information texts. It is required if *I-lines have been used in the timetable data. The text is indicated in relation to an infotext number. The infotext numbers are used in the *I-lines in order to enable the assignment of a particular text (or block of XML) to a journey.

Furthermore it is possible to define commonly valid information texts, which are not assigned to a certain station or train. A commonly valid information text entry starts with the infotext code followed by the operation day number. The operating day number "**000000**" or 6 blank characters indicate that the information text is valid daily; for all other numbers there are corresponding entries in a separate file (see file BITFELD). The entry ends with the actual commonly valid information text.

File format:

Column	Type	Meaning
$1-7$	*INT32*	Infotext number
$9ff$	*CHAR*	Train information text

Example:

```
0000001 North-Express
0000002 South-Express
```

Explanation:

To the infotext number 1 belongs this text: North-Express.

To the infotext number 2 belongs this text: South-Express.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	Infotext code
$4-9$	*INT32*	Operating day bit field number (optional)
$11ff$	*CHAR*	Commonly valid information text

Example:

```
ZN        A commonly valid information
ZN 000000 Yet another commonly valid information
ZN 081224 Merry Christmas
```

Explanation:

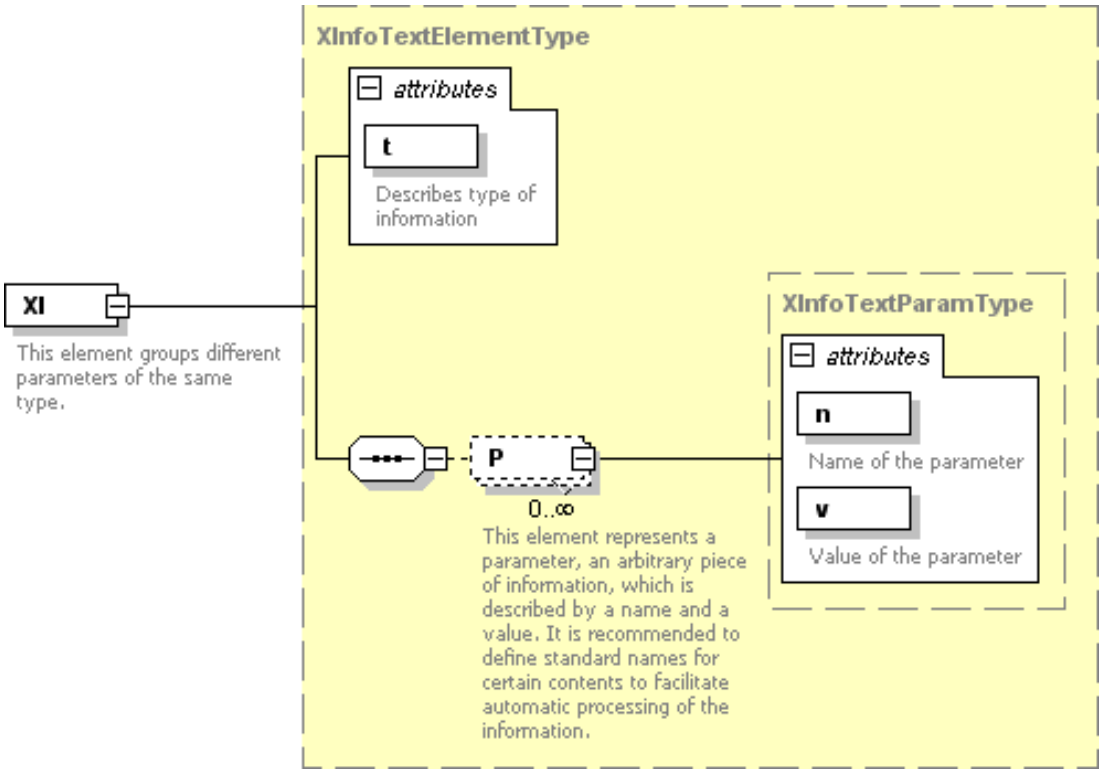
The infotext code “ZN” belongs to all 3 commonly valid information texts.

The infotext “A commonly valid information” is valid daily. The infotext “Yet another commonly valid information” is valid daily. The infotext “Merry Christmas” is valid for the operating days 081224. (see file BITFELD).

6.5.1. Extended infotexts

A special case are infotexts containing XML. Using these extended infotexts allows for better structured content, for example to enable formatted output in the interfaces.

These are designed as follows:



The exact specs of the XML format are to be found in the addendum (Section 8.1.7).

Example:

```
00000001 <XI type="Info"><P n="name" v="Börjes  ...
        ... Kiosk"/></XI><XI type ="OH"><P n="mån-fre"  ...
        ... v="8.30-20.00"/><P n="Sön" ...
        ... v="11.0018.00"/></XI><XI type="Add">...
        ... <P n="street" v="Storgatan 5"/> ...
        ... <P n="municipality" v="112 14 Stockholm"/></XI>
00000002 <XI type="Info"><P n="name" v="Parkera och åk - ...
        ... Lindholmen"><P n="desc" v="Snabbast in ...
        ... till Stockholms östra från Lindholmen är ...
        ... Roslagsbanan som tar cirka 39 minuter."/> ...
        ... </XI><XI type="Park"><P n="spaces" v="34"/></XI>
00000003 <XI type="Info"><P n="name" v="SL Center ...
        ... Täby Centrum (bussterminalen)"/> </XI>...
        ... <XI type="OH"><P n="mån-lör" v="6.30-23.15"/> ...
        ... <P n="mån-lör" v="6.30-23.15"/>...
        ... <P n="Sön" v="7.00-23.15"/></XI>
```

6.6. List of interchange stops [KMINFO]

Hafas automatically recognises the interchange points of a network. Within the file KMINFO other interchange stops can be defined. The file contains the following information for each interchange point:

- stop number
- interchange flag (numerical value)
- name of the stop

File format:

Column	Type	Meaning
$1-7$	*INT32*	the number of the stop
$9-13$	*INT16*	the value 30000 defines a stop as an interchange stop. With the value 0, a stop is blocked from interchange.
$15ff$	*CHAR*	(optional) plain text of stop name.

Example:

```
8000105  30000 Frankfurt (Main) Hbf
8000150  0     Hanau Hbf
8000152  30000 Hannover Hbf
```

Explanation:

The stops Frankfurt (Main) Hbf and Hannover Hbf are set as interchange stops. The stop Hanau Hbf is blocked for interchange.

Important: *Hafas* will only provide best performance if it calculates the interchange stops on its own. This file should never be used to specify all interchange stops since in general performance will be seriously impaired.

6.7. Specific interchange times

The specification of a minimum interchange time for a stop is often too general. For that reason, *Hafas* offers various options for specifying interchange times in detail. It is possible to enter minimum interchange times for connecting two administrations (operators), two lines' services or even two services.

6.7.1. Processing sequence of interchange times within *Hafas*

The, within the rawdata, specified interchange times are processed within *Hafas* in the following order:

1. Interchange times between services (6.7.5)
2. Line related interchange times at specificied stations (6.7.3)
   Ascending with the number of stars
3. Interchange times between administrations at specificied stations (6.7.2)
4. Stop-related interchange times (5.7)
5. Line related interchange times (globally valid) (6.7.3)
   Ascending with the number of stars
6. Interchange times between administrations (globally valid) (6.7.2)
7. Standard interchange time (5.7)

6.7.2. Interchange times between administrations [UMSTEIGV]

Interchange times between administrations make sense, if the administration codes an operator and a mean of transport. These interchange times can be defined globally and/or for individual stops.

File format:

Column	Type	Meaning
1−7	*INT32*	stop number or '@@@@@@@'
9−14	*CHAR*	administration 1
16−21	*CHAR*	administration 1
23−24	*INT16*	minimum interchange time

Example:

```
@@@@@@@ 000101 000102 02
```

Explanation:

If '@@@@@@@' is used as the stop number, the time entry applies to all stops which have not been explicitly listed. Between administration 000101 and 000102 the general interchange time for all services is 2 minutes.

6.7.3. Line and direction related interchange times [UMSTEIGL]

In this file special interchange times for the services of a line and direction to the services of another line and direction are defined. The following information is required:

- the stop number,
- the administrations,
- the means of transport,
- the line number,
- the direction flags,
- the interchange time.

In addition, the interchange may be labelled as guaranteed. Guaranteed interchanges will be preferred in the connection search of the *Hafas* system. It is possible to mark interchanges within a connection as guaranteed interchanges, too.

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number
$9-14$	*CHAR*	Administration 1
$16-18$	*CHAR*	Service category 1
$20-27$	*CHAR*	Line 1
$29-29$	*CHAR*	Direction 1
$31-36$	*CHAR*	Administration 2
$38-40$	*CHAR*	Service category 2
$42-49$	*CHAR*	Line 2
$51-51$	*CHAR*	Direction 2
$53-55$	*INT16*	Interchange time in minutes
$56-56$	*CHAR*	Exclamation mark (”‘!”’), to indicate a guaranteed interchange (optional)
$58ff$	*CHAR*	stop name (optional)

If '@@@@@@@' is used as the stop number, the time entry applies to all stops which have not been explicitly listed. The quadrupel of administration, service category/means of transport, line number, and direction flag can be filled from the right side with the wildcard character *. If instead of a direction flag a * is indicated the entry applies to all directions of the appropriate line.

If instead of a line number and a direction flag a * is indicated, the interchange time between the means of transport at the appropriate stops is defined. If only administrations are indicated the entries corresponds to the file UMSTEIGV.

If contradictory times are defined the entry with the fewest wildcard characters is valid.

It is basically possible to use quadrupel with different numbers of wildcard characters in one line. Nevertheless, any ambiguities must not occur.

Example:

```
8004440 000001 U-B 00003079 1 000002 S-B 00003189 2 002 Flittstadt
```

Explanation:

At the stop Flittstadt the interchange time between services of administration 000001, the service category U-B, the line 00003079 in direction 1 AND the services of administration 000002, the service category S-B, the line 00003189 in direction 2 IS 2 minutes.

Example:

```
@@@@@@@ 000001   *        * * 000002   *        * * 002 Flittstadt
```

Explanation:

This entry corresponds to the example of the data file UMSTEIGV.

Example:

```
8004440 000001 U-B        * * 000002 S-B 00003189 * 002 Flittstadt
8004440 000001 U-B 00003079 * 000002 S-B        * * 003 Flittstadt
```

Explanation:

These entries are contradictory. The question is: which time applies between the line number 3079 and 3189? Such entries should be clarified by an additional entry.

Example:

```
8004440 000001 U-B 00003079 * 000002 S-B 00003189 * 002 Flittstadt
```

6.7.4. Guaranteed interchanges between lines [UMSTFWL]

Similar to interchange times related to lines and directions (cf. section 6.7.3), it is possible to connect lines by guaranteed walking links. Each guaranteed walking link has to be defined within the METABHF file, the definition including a unique identifier (see section 5.6.9). Guaranteed walking links will be preferred by *Hafas* searching for connections.

File format:

Column	Type	Meaning
1−7	*INT32*	Number of the guaranteed walking link
9−14	*CHAR*	Administration 1
16−18	*CHAR*	Service category 1
20−27	*CHAR*	Line 1
29−29	*CHAR*	Direction 1
31−36	*CHAR*	Administration 2
38−40	*CHAR*	Service category 2
42−49	*CHAR*	Line 2
51−51	*CHAR*	Direction 2

Example:

```
1234567 000001 U-B 00003079 1 000002 S-B 00003189 2
```

Explanation:

The guaranteed walking link 1234567 is applicable to interchanges between services of administration 000001, category U-B, line 00003079, direction "'1"' and services of administration 000002, category S-B, line 00003189, direction "'2"'.

Example:

```
7654321 000001   *        * * 000002   *        * *
```

Explanation:

The guaranteed walking link 7654321 is applicable to interchanges between services of administration 000001 and services of administration 000002.

6.7.5. Interchange times between services[UMSTEIGZ]

If two trains stop at the same platform, the minimal interchange time for this stop will normally be underrun. With an entry in the UMSTEIGZ file it becomes possible for the program to find this possibility to change anyway. In this file specific interchange times between two services can be defined. The following information is required:

- stop number
- identification of the first service (service number and administration)
- identification of the second service (service number and administration)
- interchange time in minutes
- stop name to improve readability

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number
$9-13$	*INT32*	Service number 1 (5-digit)
$15-20$	*CHAR*	administration for service 1
$22-26$	*INT32*	service number 2
$28-33$	*CHAR*	administration for service 2
$35-37$	*INT16*	interchange time in minutes
$38-38$	*CHAR*	Exclamation mark (”‘!”’) to indicate a guaranteed change (optional)
$40ff$	*CHAR*	stop name (optional for readability)

In the connection search of the *Hafas* system, guaranteed interchanges will be preferred. Guaranteed interchanges which occur within the resulting connections may be marked.

Example:

```
8002010   3079 80____ 3189 80____ 002 Flieden
```

Explanation:

At the stop Flieden, the interchange time between service 3079 of administration 80 and service 3189 of administration 80 is 2 minutes, irrespective of the time defined in the file for the station-related interchange times. You can use this line to encode the following information:

- Interchange allowed: if the difference between arrival timeof service 3079 and departure time of service 3189 according to the timetable is greater than or equal to two minutes, the interchange is allowed.
- interchange not allowed: if the difference according to the timetable is less than two minutes the interchange will not take place. It is therefore possible to block the interchange between two services.

6.7.6. Guaranteed walking links between services [UMSTFWZ]

Similar to interchange times related to services (cf. section 6.7.5), it is possible to connect services by guaranteed walking links. Each guaranteed walking link has to be defined within the METABHF file, the definition including a unique identifier (see section 5.6.9). Guaranteed walking links will be preferred by *Hafas* searching for connections.

File format:

Column	Type	Meaning
$1-7$	*INT32*	Number of the guaranteed walking link
$9-13$	*INT32*	Service number 1 (5-digit)
$15-20$	*CHAR*	Administration for service 1
$22-26$	*INT32*	service number 2
$28-33$	*CHAR*	Administration for service 2

Example:

```
1234567  3079 80____  3189 80____
```

Explanation:

The guaranteed walking link 1234567 connects service 3079 of administration 80____ to service 3189 of administration 80____.

6.8. Associations [VEREINIG]

The term “association” means the combining of two different services for one specific route section. This file indicates that two services run as one physical unit and an interchange process may take place onboard.

The file contains:

- stop number of first stop of the common section
- stop number of last stop of the common section
- service number and administration number service 1
- service number and administration number service 2
- stop names

Per line are indicated: the two stops, between which the services are combined, the services themselves with service number and administration. Optional the stop names can follow. A line is interpreted as follows: The services F1 and F2 in the section between stop H1 and H2 are associated.

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number 1
$9-15$	*INT32*	Stop number 2
$17-21$	*INT32*	Service number 1
$23-28$	*CHAR*	Administration for service 1
$30-34$	*INT32*	Service number 2
$36-41$	*CHAR*	Administration for service 2
$43ff$	*CHAR*	Optional comment

Example:

```
8000128 8000152 3062 DB0004 3082 DB0004 Goettingen - Hannover
```

Explanation:

Between the stops Göttingen (8000128) and Hannover (8000152) service 3062 of administration DB0004 and service 3082 of administration DB0004 are associated as service 3062.

6.9. Continuous operation [DURCHBI]

A continuous operation is given if two independent services are operated after each other by the same trainset so that physically one continuous service for the passenger results (kind of association).

The following definitions are used for representation:

- service number and administration of service 1
- stop number at which service 1 ends
- service number and administration of service 2
- operating days on which the continuous operation takes place
- stop number at which service 2 starts
- attribute to mark the continuous operation

A line is interpreted as follows: Service 1 continues as service 2. This applies on all the operating days specified. Using the attribute it is possible to identify the stops of the continues operation.

File format:

Column	Type	Meaning
$1-5$	*INT32*	Service number 1
$7-12$	*CHAR*	Administration for service 1
$14-20$	*INT32*	Last stop of service 1
$22-26$	*INT32*	Service number 2
$28-33$	*CHAR*	Administration for service 2
$35-40$	*INT16*	Operating day bit field number
$42-48$	*INT32*	First stop of service 2 (optional)
$50-51$	*CHAR*	Attribute code to mark the continues operation (optional)
$53ff$	*CHAR*	Comment after “%” (optional)

If the first stop of service 2 is missing then service 2 will start at the last stop of service 1.

Example:

```
08844 DB0003 0000253 03192 DB0002 000001 Mönchengladbach Hbf
```

Explanation:

Service 8844 (administration DB0003) runs from Mönchengladbach as service 3192 (administration DB0002). The continuous service exists on all operating days specified by the operating day code 000001.

6.10. File with direction information [RICHTUNG]

If the direction of a service cannot be given as a stop in its line of route, it can be defined in this file.

File format:

Column	Type	Meaning
$1-7$	*CHAR*	direction code
$9ff$	*CHAR*	direction information in plain text (max. 50 characters)

Example:

```
1111111 Hauptbahnhof/ZOB
```

Explanation:

The direction entry 1111111 in the timetable data is to be interpreted as “Direction Hauptbahnhof/ZOB” (Central Station/Central Bus Terminal).

6.11. File with border stop information [GRENZHLT]

If a border stop in the timetable data does not appear in the stop file BAHNHOF, an entry must be present in this file.

File format:

Column	Type	Meaning
$1-7$	*CHAR*	Border stop code
$9ff$	*CHAR*	border stop name in plain text (max. 50 characters)

Example:

```
1111111 border stop Passau
```

Explanation:

The border stop 1111111 appears in the timetable data and is not a stop in the stop file BAHNHOF. It is interpreted as “Border stop Passau”.

6.12. Time displacement file [ZEITVS]

This file contains specifications for time displacements of the stops. The file ZEITVS replaces the older file ZEITZONE. From it, the binary time change file PLANZZ is created.

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number
$9-13$	*INT32*	Time displacement (+HHMM, +HH, -HHMM or -HH)
$15-22$	*INT32*	Starting date (DDMMYYYYor DDMMYY)
$24-27$	*INT16*	Starting time (HHMM) (optional)
$29-36$	*INT32*	ending date (DDMMYYYY or DDMMYY)
$38-41$	*INT16*	ending time HHMM (optional)
$43ff$	*CHAR*	comment after “%” (optional)

alternatively:

File format:

Column	Type	Meaning
$1-7$	*INT32*	Stop number 1
$9-15$	*INT32*	Stop number 2
$16ff$	*CHAR*	comment after “%” (optional)

- Stop numbers must be defined as 7digit figures, if missing the definition of the preceding number applies.
- Time displacements must start with a “+” or “-”; values between -14 and +14 are interpreted as hours, other values as hours and minutes.
- Time specifications after the dates are optional. Missing definitions are assumed as to be 0000 for the first day and 2400 for the last day specified.
- A missing ’starting date’ is interpreted as the begin of the timetable period and a missing ’ending date’ as the end of the timetable period.

- A stop number can get various time displacements and dates/times in the same line.
- Later created time displacements overwrite earlier created ones (in case there is a time overlap).
- If a stop number has no time displacement with a date combined, the data of the last valid preceding line is used.
- Instead of time displacements with date a second stop number can be indicated, both stops are interpreted identical regarding their time displacements (changes affect both stations).
- Stop numbers without time displacement obtain the time displacement of the last preceding stop number.

It should be paid attention for only using one of the following notations. Every new time displacement starts with the specified column 9 - 12 (e.g. +2).

The example from the ZEITZONE-file can be expressed as a ZEITVS-file this way:

Example:

```
0000000 +2 27051993 +1 30091993 +2 31031994 % local transport
1000000 +3 27051993 +2 30091993 +3 31031994 % Finland
7000000 +1 27051993 +0 28101993 +1 31031994 % Great Britain
8000000 0000000     % DB as local transport
```

Explanation:

In the local transport a time displacement for 2 hours effective from 27/05/93 is planned. Effective from 30/09/93 1 hour and from 31/03/94 again 2 hours.

The stop number range from 8000000 is puit on a par with the specifications from 0000000.

As an alternative the following style would be possible:

Example:

```
0000000 +1      % local transport standard time
1000000 +2      % Finland standard time
...
```

... or further down or in a file 'summer time':

```
 1000000 +3 30031993 0000 30091993 0200            % Finland summer 1993
         +3 31031994 0200 29091994 0200            % Finland summer 1994
...
```

Explanation:

In the first part the basic times for the countries are defined (Finland +2 hours, ...). Following, exceptions are defined.

Finland summer:

From 27/05/93 at 02:00 o'clock +3 hours is valid until 30/09/93 at 02:00 o'clock.

From 31/03/94 at 02:00 o'clock +3 hours is valid until 29/09/94 at 02:00 o'clock

Example:

```
1000000 +3 27051993 0200 +2 30091993 0200 +3 31031994 0200 % Finland summer 1993
```

Explanation:

From 27/05/1993 02:00 o'clock applies +3 hours, from 30/09/1993 02:00 o'clock applies +2 hours, from 31/03/1994 02:00 o'clock applies +3 hours. However one should use only one line style, not three different as in the last part of the example.

6.13. Exchange file [EXCHANGE]

The exchange file EXCHANGE enables mapping of stops onto other stops. This file is useful if the same stop has been given different numbers in different data sets. The following definitions are used for representation:

- the stop numbers of the stops to be mapped,
- the stop name

File format:

Column	Type	Meaning
$1-7$	*INT32*	stop number 1
$9-15$	*INT32*	stop number 2
$17ff$	*CHAR*	(optional) stop name

Example:

```
8503052 8503053 Schweighof
```

Explanation:

Each occurrence of the stop 8503052 is replaced by the stop 8503053.

6.14. Address and building file [ADRESSEN]

In the local transport version of *Hafas* it is also possible to enter an address as start or destination of a trip. Therefor it is necessary to make an assignment between stops and addresses. This assignment is undertaken in the file described.

The following definitions are used for representation:

- the stop number
- distance between stop and building
- street or building name
- the stop name
- the post code

File format:

Column	Type	Meaning
$1-7$	*CHAR*	stop number
$9-12$	*INT16*	distance stop - address (building) in multiples of 10 m
$14-43$	*CHAR*	house number
$45-74$	*CHAR*	street name or building name
$76-85$	*CHAR*	post code
$87ff$	*CHAR*	(optional) stop name

If a house number is entered, this entry relates to an address. If the house number is missing (blank), this is an entry for a building.

Example:

```
8500123 0005     Abonnementsbüro VBZ   Bahnhofplatz
0100351 0020 1A  Aargauerstrasse       CH-1000
0100000 0021 1A  Aargauerstrasse       CH-1000
```

Explanation:

The Abonnementsbüro (ticket office) VBZ (post code not specified) is 50 metres from stop 8500123 (Bahnhofplatz). The address Aargauerstrasse 1A (post code “CH-1000”) is 200 metres from stop 0100351 and 210 metres from stop 0100000.

6.15. Sort file [SORTKEYS]

The stop names are sorted according to the entries in this file. A sorting code is assigned to each representable character. All characters are arranged according to the size of their sort code. Characters not present are placed at the end of the selection list.

The smaller the sort code, the more ahead the stop name is in the list.

File format:

Column	Type	Meaning
$1-1$	*CHAR*	character (code between 0 and 255)
$3ff$	*CHAR*	sorting code (0 to 29999)

Example:

```
100
% do not differentiate between upper/lower case letters.
% umlauts, accents
=====================================================
A 200
a 200
â 200
ä 200
à 200
å 200
á 200
Ä 200
Å 200
```

Explanation:

Blank stands in front of all other characters. All the letters: A, a, â, ä, à, å, á, Ä and Å are treated equally.

6.16. Additional stop information [BFINFO]

Sometimes it is desirable to provide additional information on selected stops.

Example 1: Connecting services run from a stop to a number of surrounding villages. This information can be represented. Example 2: There are destinations which cannot be reached with public services. It is possible, for example, to connect Euro-Disneyland with Paris and to store for Euro-Disney further information on the route Paris$->$Euro-Disneyland.

The data records are identified by the following entries:

File format:

Column	Type	Meaning
$1-1$	*CHAR*	fixed character “$”
$2-8$	*CHAR*	stop number
$10ff$	*INT16*	optional text

The text may comprise as many lines as required to be output as information. It is possible to arrange data information logically for the representation by inserting lines which consist only of the text “@@EOB”. At these points, the displaying program breaks down the text automatically.

Example:

```
$8000298     Passau HbF
Bad Füssing: bus connection 5 km from Pocking
Bad Füssing: bus connection 30 km from Passau
Through tickets available!
```

Explanation:

At stop “Passau Hbf” (8000298) this text:

“ Bad Füssing: bus connection 5 km from Pocking

Bad Füssing: bus connection 30 km from Passau

Through tickets available!”

is printed out.

6.17. Platform/bus shelter information [GLEISE]

In this file it is possible to provide platform or bus shelter information for each service.

File format:

Column	Type	Meaning
1−7	*INT32*	stop number
9−13	*INT32*	service number
15−20	*CHAR*	administration for service
22−29	*CHAR*	platform information (left aligned!)
31−34	*INT16*	optional: time in the format HHMM (¡2400)
36−41	*INT32*	optional: operating day code

Thus a platform name can be assigned to each stop of a service.

The specification of the time is necessary if a vehicle uses different platforms at the same stop (e.g. arrival platform 3, departure platform 5) within one service. In this case, the correct information can be identified on the basis of the time. Normally the time field remains empty.

The operating day code is usually 000000 (or missing): each time the service perform the stop the same platform is used. If the platforms differ on specific days, however, (e.g. weekdays platform 4, sundays platform 2), the data set has to be recorded several times and the valid information identified by means of the operating day index number.

Example:

```
8010338 00319 DB0074 6              100201
8010338 00319 DB0074 5              100202
8010338 03232 DB0011 3
```

Explanation:

All platform information apply to the stop 8010338 (Stralsund). The service 00319 of administration DB0074 calls on the platform 6 on the days specified with the bitfield no.100201. On other days specified with no. 100202 the same service calls on platform 5.

The service 03232 of administration DB0011 calls on platform 3 for all the days the service operates.

6.18. Operator information [BETRIEB]

By means of the file BETRIEB different administrations can be associated with one operator. Each operator has an abbreviated name (max. three characters), a long name (max. eight characters) and a full name (arbitrary length). The purpose is to be able to associate each service with its operator.

Each administration may be associated to only one operator. Administrations not listed in the file are associated to the operator number 00000. It is possible to assign names for this operator number in the file BETRIEB.

Structure of the lines

Each line begins with a operator number. The number may not exceed the value 32767. The following types are allowed:

Type	Meaning
K	abbreviated name (max. 3 characters)
L	long name (max. 8 characters)
V	full name
:	list of administrations

The names may contain blanks. In this case they have to be enclosed in ’ ’ or " ". The inverted commas do not belong to the name. If the characters ’ or " are a part of the name they have to be enclosed by the opposite kind of character (" or ’).

A list of administrations may occur only at the end of a line. It is possible to specify several lines with lists of administrations for the same operator.

Example:

```
00001 K DB L ’DB AG’ V ’Deutsche Bahn AG’
00001 : 80____ 80a___ 80b___
```

Explanation:

In this example the operator 00001 with the abbreviated name *DB*, the long name *DB AG* and the full name *Deutsche Bahn AG* is defined. The Deutsche Bahn AG comprises the administrations 80____ , 80a___ and 80b___.

6.19. Extra trains [SONDERZG]

With the *Hafas* algorithm it is possible not only to find the fastest service but also more convenient services which involve fewer interchanges. For example services which include relief trains are not found if these are slower than or even overtaken by their corresponding trains. In the file SONDERZG pairs of regular services and relief services are indicated, which are used by *Hafas* to find the timetable information desired.

Structure of the lines

Each line contains one pair of trains specified with administration and train number.

File format:

Column	Type	Meaning
1−5	*INT32*	Service number 1
7−11	*CHAR*	Administration of service number 1
13−17	*INT32*	Service number 2
19−23	*CHAR*	Administration of service number 2

Every line consists of one service pair, where the services are indicated by administration and service number.

Example:

```
00001 80____ 00002 81____
```

Explanation:

The service 00001 of the administration 80____ is a regular service to the relief service 00002 81____.

6.20. Address fine dissolution[address.txt]

Characteristic data of whole streets, the sections of streets or single addresses are needed for the address fine dissolution.

Type	Meaning
CHAR	Postal code
CHAR	Place name
CHAR	Street name
CHAR	First house number left roadside
CHAR	Last house number left roadside
CHAR	First house number right roadside
CHAR	Last house number right roadside
FLOAT	X-coordinate starting point
FLOAT	Y-coordinate starting point
FLOAT	X-coordinate destination
FLOAT	Y-coordinate destination
INT16	Counting method of the house numbers on the left road side
INT16	Counting method of the house numbers on the right road side

The columns have to be separated by a semicolon (`;`).

Empty columns are permitted (`;;`). Further on applies:

- The postal code may be empty.
- Place name or street name may be empty, but not both.
- The house numbers may be completely or partially empty. So streets without known house numbers, single addresses (only first house number left roadside) or sections of streets with house numbers on one side (left road side) only can be displayed.
- For the coordinates a geodecimal (WGS84) or a planar coordinate system (kilometer precision) may be used
- The coordinates for the starting point have to be named.
- The columns with coordinates for the destination and with the counting methods may be empty or completely missing.

- For the counting methods applies:
  - 1 is for the odd house numbers
  - 2 is for the even house numbers
  - 3 is for the continous house numbers

Example:

```
30163;Lister Str.;Hannover;35;34;16;18;9.7;52.3;9.7;52.3;3;3
;Lister Str.;Hannover;35;34;;;9.75111000;52.39091100;;;;
;Lister Str.;Hannover;35;;;;9.75111000;52.39091100;
```

Explanation:

- The first line includes the maximal quantity of informations for one section of a street. The house numbers on both road sides are continous.
- In the second line some informations are missing. Only one road side has house numbers. This street section has only coordinates for the starting point.
- The third line is only a single address. The last four not used columns have been omitted.

6.21. Coordinates for house numbers, street sections, and crossings [HAUSNR]

For an address pool which consists of streets, it is possible to define several single addresses, street sections, and crossings. Each housenumber, each street section, and each crossing must be defined by a block of two or three lines:

File format:

Column	Type	Meaning
1−7	*INT32*	Number of the street in the address pool
9−18	*FLOAT*	x-Coordinate of the starting point of the street section, of the address, or of the crossing
20−29	*FLOAT*	y-Coordinate of the starting point of the street section, of the address, or of the crossing
31−40	*FLOAT*	x-Coordinate of the end point of the street section (street section only)
42−51	*FLOAT*	y-Coordinate of the end point of the street section (street section only)

For the coordinates a geodecimal (WGS84) or a planar coordinate system (kilometer precision) may be used

The coordinate line must be followed by a line with a single address:

File format:

Column	Type	Meaning
1−2	*CHAR*	**H*
4−8	*CHAR*	house number

by one or two lines with one or two street sections:

File format:

Column	Type	Meaning
$1-2$	*CHAR*	**A*
$4-4$	*INT16*	count type for the house numbers
$6-10$	*CHAR*	first house number of the street section
$12-16$	*CHAR*	last house number of the street section

or by one line with a crossing:

File format:

Column	Type	Meaning
$1-2$	*CHAR*	**C*
$4-10$	*INT32*	Number of the crossing street in the address pool. Can be filled with @ if the number is unknown or if the crossing street i no in the address pool
$12ff$	*CHAR*	Name of the crossing street

For single house numbers an alphanumerical add on can be given. For street sections only numerical values are allowed.

The count type can be

- 1 for odd house numbers.
- 2 for even house numbers.
- 3 for continous house numbers.

It is not necessary to define the number of a crossing street, but if defined it can be used to save space.

Example:

```
9900000 10.4670780 53.2678300 10.4627700 53.2698480
*A 3 1     3
9900000 10.4383335 53.2793750
*H 4
9900000 10.4383335 53.2793750
*H 4a
9900000 10.4383335 53.2793750
*H 4b
9900000 10.4520070 53.2888410 10.4590190 53.2894900
*A 1 5     9
*A 2 6     14
9900000 10.4520070 53.2888410
*A 5a
9900000 10.4520070 53.2888410
*C 9900001 Camp Street
9900000 10.4520070 53.2888410
*C @@@@@@@ Ring Street (north)
9900000 10.4520070 53.2888410
*C @@@@@@@ Ring Street (south)
```

Explanation:

The street with the number 9900000 has street sections with house number from 1 to 3 (continous), from 5 to 9 (odd), from 6 to 14 (even), and single house numbers 4, 4a, 4b, and 5b. The street has crossings with the Camp Street and the Ring Street (north and south).

6.22. Virtual borders[SPERRKANTEN]

In this file are the borders, which are the base for the virtual border routing.

File format:

Column	Type	Meaning
1−10	*FLOAT*	x-coordinate starting point
12−21	*FLOAT*	y-coordinate starting point
23−32	*FLOAT*	x-coordinate destination
34−43	*FLOAT*	y-coordinate destination

For the format of the coordinates there are two possibilities:

1. The coordinates of the belonging plan data - pool are in a planar format: In this case the coordinates of the borders have to be kilometer-precise in the same planar format.
2. The coordinates of the belonging plan data - pool are geodecimal coordinates (WGS84).

Example:

```
514.985000 4454.58100 514.909000 4454.421000
```

Explanation:

The border runs from (514.985; 4454.581) to (514.909; 4454.421).

6.23. Road points [STRECKENPT]

This file contains points in the route network which are no stops and additional realgraph information on stops.

File format:

Column	Type	Meaning
1−7	*CHAR*	road point short name or external stop number
9−18	*FLOAT*	x-coordinate
20−29	*FLOAT*	y-coordinate
31−35	*INT16*	z-coordinate (optional)
37−86	*CHAR*	road point identifier or stop name

Each road point short name must not be identical to an external stop number. The coordinates of stops may differ from the coordinates defined in the stop coordinates file BFKOORD. Adding stops which are used by services to the road points file is optional.

Example:

```
POINT01 9.71451865 46.6531972 100   Point 1
```

6.23.1. *I-Line (optional)

A *I-line assigns an infotext to a road point. Refer to section 6.5 for more information on infotexts.

File format:

Column	Type	Meaning
1−2	*CHAR*	*I
4−5	*CHAR*	Infotext code
7−13	*INT32*	Infotext number

A road point can have multiple infotexts. The infotext code must not contain whitespace characters. The infotext code creates a classification or grouping of infotexts. Infotext codes consisting of two capital letters have a defined interpretation. Other infotext codes can be used without restrictions.

Example:

```
POINT01 9.71451865 46.6531972 100   Point 1
*I bc 1234567
*I cd 1234568
8000152 9.742022   52.377192  55    Hannover Hbf
*I AB 1234569
```

Explanation:

This example defines a road point POINT01 having the realgraph infotexts 1234567 and 1234568. The realgraph infotext 1234569 is assigned to the stop 8000152.

6.23.2. **M*-Line (optional)

A **M*-line contains a list of road point marks. These road point marks can be characterized as flags for realgraph road points. A road point mark consists of a single non-whitespace character. Capital letters have a defined interpretation. All other non-whitespace characters can be used without restrictions. The interpretation of the characters is not deposited in the rawdata. It is done by HAFAS.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	**M*
$4ff$	*CHAR*	road point mark list

There must not be more than one **M*-line for a road point.

Example:

```
POINT01 9.71451865 46.6531972 100   Point 1
*M abE12
```

Explanation:

The example defines a road point POINT01. The road point has the road point marks “a”, “b”, “E”, “1” and “2”. The interpretation of the capital letter “E” is defined as “is electrified”.

6.24. Realgraph edges [KANTEN]

The realgraph reflects the route network. This file contains the edges of the realgraph.

File format:

Column	Type	Meaning
1−8	*CHAR*	node 1: stop number or road point short name
10−17	*CHAR*	node 2: stop number or road point short name
19−19	*CHAR*	B for bidirectional edges (optional)

A row creates an edge from node 1 to node 2. If there is a “B” attached an edge from node 2 to node 1 is created too.

Example:

```
8501566 8501564 B
```

Explanation:

Adds the edges (8501566, 8501564) and (8501564, 8501566).

6.24.1. *G-lines (optional)

Each edge can have a sequence of intermediate nodes. Each intermediate node must be placed along at most one edge.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	**G*
$4ff$	*CHAR*	intermediate nodes, distances, marks and infotexts

There may be several **G*-lines.

Between two nodes, before the first intermediate node and after the last intermediate node there can be

- an optional “L” followed by the distance of the nodes in metres,
- an optional “M” followed by a list of edge marks and
- an optional “I” followed by a two character infotext code and an infotext number (multiple “I”-blocks are possible).

If no distance is given the distance is set to the air-line distance. For more information on edge marks refer to section 6.24.5. Infotexts are described in section 6.5. Refer to section 6.24.4 for more information on infotexts for real-graph edges.

Example:

```
8501566 8501564
*G L 450 8501565 SIGNAL7
*G L 1580 I de 1234567 I fg 1234568 P123456 M AB12
```

Explanation:

The example defines an edge between the nodes 8501566 and 8501564 via the nodes 8501565, SIGNAL7 and P123456. The length of the edge from node 8501566 to node 8501565 amounts to 450 m, the length of the edge from node SIGNAL7 to node P123456 amounts to 1580 m. The lenghts of the edges (8501565, SIGNAL7) and (P123456, 8501564) are set to the air-line distance because no distance is specified. The infotexts 1234567 and 1234568 are attached to the edge (SIGNAL7, P123456). The edge (P123456, 8501564) has the edge marks “A”, “B”, “1” and “2”.

6.24.2. *L-Line (optional)

If the total length of an edge is known but not all lengths of the sub edges then it's possible to define the total length using a *L-line. If there is no *L-line but intermediate nodes the distance is set to the sum of the distances of the intermediate nodes otherwise the distance is set to air-line distance. The sum of the distances of the intermediate nodes must not be greater than the total distance.

File format:

Column	Type	Meaning
1−2	*CHAR*	*L
4−13	*INT32*	edge length in metres

There must be at most one *L-line.

Example:

```
8530260 8504316 B
*L 500
```

Explanation:

The example defines an edge between the nodes 8530260 and 8504316. The length of this edge amounts to 500 metres.

6.24.3. *T-Lines (optional)

The edges of the realgraph are often only appropriate for certain means of transport or they are only used by certain means of transport. *T-lines determine by which means of transport an edge may be used. The means of transport are defined based on service categories and product classes. Furthermore edges may be defined as footpath edges. Refer to section 5.5 for more information on service categories and product classes. For product classes the subsequent line format is used:

File format:

Column	Type	Meaning
1−3	*CHAR*	*TP
5−6	*CHAR*	product class (from 0 to 13)

The line format for service categories is

File format:

Column	Type	Meaning
1−3	*CHAR*	*TG
5−7	*CHAR*	service category code

The subsequent table lists the line format for footpaths:

File format:

Column	Type	Meaning
1−3	*CHAR*	*TF

A continuous block of *T-lines defines the permissible service categories for the subsequent edges. A new *T-line block overrides service category settings from previous *T-line blocks. Edges listed up to the first *T line may be used by any service category but may not be used for footpaths. The different *T-line types may be combined arbitrarily. For edges resulting from intermediate nodes from a *G-line the service types of the corresponding “main” edge are used.

Example:

```
8530260 8504316 B
*TP 02
*TG IC
*TG ICE
8530261 8504370 B
*TG M
8530262 8504371 B
8504371 8504372 B
*TG BUS
*TF
0123456 0123457 B
```

Explanation:

The edge (8530260, 8504316) may be used by any service category. The usage of the edge (8530261,8504370) is permitted for all service categories having product class 02 and for the service categories “IC” and “ICE”. The edges (8530262, 8504371) and (8504371, 8504372) may be used by services having category “M”. The edge (0123456, 0123457) is enabled for the service category “BUS” and for footpaths.

6.24.4. *I-Line (optional)

A *I-line assigns an infotext to an edge. Refer to section 6.5 for more information on infotexts.

File format:

Column	Type	Meaning
1−2	*CHAR*	*I
4−5	*CHAR*	Infotext code
7−13	*INT32*	Infotext number

An edge can have multiple infotexts. The infotexts are also assigned to all edges defined by *G-lines dedicated to the edge. The infotext code must not contain whitespace characters. The infotext code creates a classification or grouping of infotexts. Infotext codes consisting of two capital letters have a defined interpretation. Other infotext codes can be used without restrictions.

Example:

```
8530260 8504316 B
*I bc 1234567
*I cd 1234568
```

Explanation:

This example defines an edge from node 8530260 to node 8504316 having the infotexts 1234567 and 1234568.

6.24.5. *M-Line (optional)

A *M-line contains a list of edge marks. These edge marks can be characterized as flags for edges. An edge mark consists of a single non-whitespace character. Capital letters have a defined interpretation. All other non-whitespace characters can be used without restrictions. The interpretation of the characters is not deposited in the rawdata. It is done by HAFAS.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	*M
$4ff$	*CHAR*	Edge mark list

There must not be more than one *M-line for an edge. The edge marks are also assigned to all edges defined by *G-lines dedicated to the edge.

Example:

```
8530260 8504316 B
*M abE12
```

Explanation:

The example defines an edge between the nodes 8530260 and 8504316. The edge has the edge marks “a”, “b”, “E”, “1” and “2”. The interpretation of the capital letter “E” is defined as “is electrified”.

6.25. Service-route-network-mapping [FAHRTZUORDNG]

This file maps services to the realgraph. It is used to define the course of the service in detail. The mapping is based on stops and road points (section 6.23). The edges file must contain appropriate edges (section 6.24).

File format:

Column	Type	Meaning
1−5	*INT32*	service number
7−12	*CHAR*	administration
14−19	*CHAR*	operating day code

The line maps the subsequent data to a service. The data applies for the specified operating days.

Example:

```
00011 000005 000123
```

Explanation:

The subsequent data refers to the service having number 00011 and administration 000005 for the operating days having code 000123.

6.25.1. **P*-lines

A **P*-line defines nodes, which are passed by the service. The data refers to the previous service line. The first node of a **P*-line must be a route stop. A service may have more than one **P*-line. The order of the **P*-lines must follow the order of the stops given by the route of the service. Sections of the route may be omitted.

File format:

Column	Type	Meaning
$1-2$	*CHAR*	**P*
$4ff$	*CHAR*	node identifier list

Within a row there may be further route stops which follow directly in the route. Intermediate nodes which are no route stops may be omitted. For each intermediate node the previous and the next stop must be listed, otherwise the intermediate node will be neglected. The route between the given nodes is calculated as the shortest path between the nodes.

Example:

```
*P 8530260 POINT09 SIGNAL7 8504316
```

Explanation:

The service passes the nodes 8530260, POINT09, SIGNAL7 and 8504316. The node 8530260 is a route stop.

6.26. Regions [REGION]

The regions file defines areas. Regions are defined by additive and subtractive polygons. A point is within a polygon if it is within at least one additive polygon and if it is not within a subtractive polygon.

File format:

Column	Type	Meaning
1−2	*CHAR*	**R*
4−9	*INT32*	region number
11−60	*CHAR*	region name (optional)

The region number is used for referencing the region in other raw data files.

Example:

```
*R 123456 Region A
```

Explanation:

This example generates a region having number 123456 and name “Region A”.

6.26.1. **P*-line

A **P*-line starts a new polygon. There are two polygon types: additve and subtractive polygons. Additive polygons generate new parts of the region, subtractive polygons remove parts of the region.

File format:

Column	Type	Meaning
1−2	*CHAR*	**P*
4−4	*CHAR*	“+” for additve polygons, “-” for subtractive polygons

The subsequent lines which do not start like **...* contain the coordinates of the edges of the polygon.

6.26.2. Polygon vertices

Lines which do not start like *... provide vertices for the current polygon. Each polygon must consist of at least three vertices. Each vertex must be placed in a separate line.

File format:

Column	Type	Meaning
1−10	*FLOAT*	x-coordinate
12−21	*FLOAT*	y-coordinate

For more information on the coordinate system used refer section 5.2. It is to be used the same coordinate system for coordinates of stops and region edges.

Example:

```
*R 1234567 Region A
*P +
8.75 47.35
8.65 47.40
9.60 46.30
*P -
8.80 47.65
8.67 47.1
9.95 46.75
*P +
8.15 47.20
8.15 46.10
9.75 46.35
```

Explanation:

This example generates a region having number 123456. The region is based on three polygons. At first the first and the last polygon are added. After that the second polygon is subtracted.

7. Explanations on file description

The following conventions for the definition of the respective data type have been used for the description of the file structure:

- **Comment lines** are identified by a “%” in the first column. They can appear as required in the files and are skipped.
- The data fields identified by **INT16** must conform to the following format (the order is important):
  - leading blanks and/or tabs
  - an (optional) preceding sign
  - sequence of numerals as required up to the first non-numeric character
  - the specified number must not fall short of or exceed the value range -32767 to 32767.
- The data fields identified by **INT32** must conform to the following format (the order is important):
  - leading blanks and/or tabs
  - an (optional) preceding sign
  - sequence of numerals as required up to the first non-numeric character
  - The specified number must not fall short of or exceed the value range -2147483648 to 2147483647.
- The data fields identified by **FLOAT** must conform to the following format (the order is important):
  - leading blanks and/or tabs
  - an (optional) preceding sign
  - sequence of numerals as required, a point and a further sequence of numerals as required,
  - an (optional) exponent introduced by e or E which includes an (optional) preceding sign as well as an (obligatory) integer value.
  - The specified number must not fall short of or exceed the value range 3.4E-38 to 3.4E+38.
- The data fields identified by **CHAR** can contain as many characters as required (IBM PC code between 0 and 255).

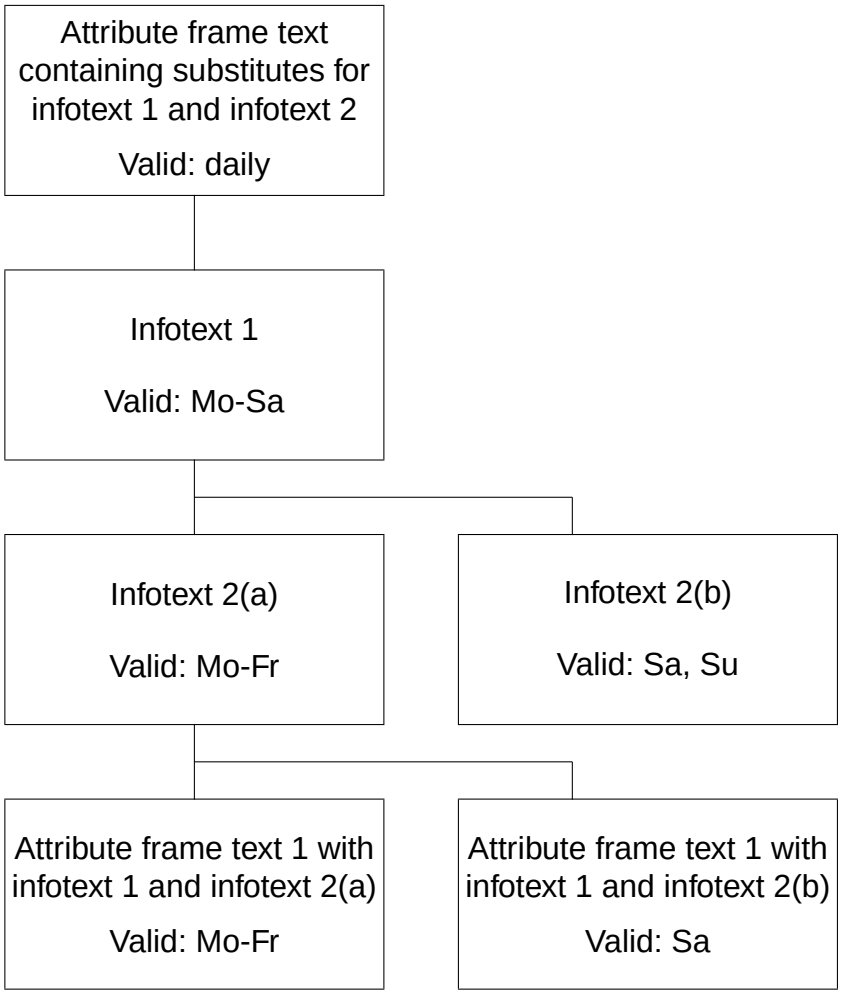
- Station, stop and bit field numbers must always be specified with leading zeros.
- All columns not described in detail (table lines in which the meaning is not explained) may contain blanks.

8. Appendix

8.1. Generation of information on stops

Texts for attributes of train services will be displayed with their valitidity and in the selected language. Attributes for stops can be displayed in the same manner. Additionally it's possible to define substitutes to enable the operation of variable text output related to a restricted validity. Since the bitfields will be adjusted to a new validity the amount of possible infotexts for stops can reach any number.

This paragraph should point out the general operation of *Hafas*



For the validity in this example generated attribute texts applies:

(validity attribute frame text) & (validity info text 1) & (validity info text 2a) = (validity attribute text 1)

(validity attribute frame text) & (validity info text 1) & (validity info text 2b) = (validity attribute text 2)

In theory, if none of the cut bit fields results 0, n^n attribute texts can be created from n info texts, which in practice probably is not necessary.

8.1.1. Relations between different rawdata files

Resembling the display of attributes for train services the attributes for stops offer a more sophisticated function: it’s possible to define an attribute frame text , which holds the basic text string and substitutes. These substitutes can be replaced with text information which may be restricted to different validity.

This feature enables the display of various texts fromout a text frame.

8.1.2. Example: telephone number for porter service

As an example we show the combined display of an attribute text and two infotexts with different validities.

The task is to advertise the proper telephone number of the porter service in Hannover Hbf.

The basic text string for the attribute GS may be as follows:

```
Gepäckträgerservice Telefon: $IGT Fax $IGF.
```

The related line in file BHFATTR is:

```
GS 300 10 Gepäckträgerservice Telefon: $IGT Fax: $IGF
```

The link into the file BHFART should read:

```
8000152 A GS 000000
```

In giving this definition the presence of the related infotext codes GT (Gepäckträgerservice-Telefon) and GF (Gepäckträgerservice-Fax) must be ensured.

The related lines in file BHFART are:

```
8000152 I GT 0002350 000000
8000152 I GF 0002351 000000
```

The bitfield number 00000 (daily) is given, which can be omitted in **A* and **I*-lines.

The final texts for display are listed in the file INFOTEXT. For our example the lines must read:

```
0002350 0511/123456
0002351 0511/654321
```

After a successful transform process *Hafas* will display the following combined text for the stop Hannover Hbf:

Gepäckträgerservice Telefon: 0511/123456 Fax: 0511/654321 (daily)

8.1.3. Example: opening hours of booking offices (multilingual)

This example shows the combination of an attribute text frame (german/english) with two infotexts which vary in their validity.

The attribute text frame for the attribute Fahrkartenschalter FS in german reads:

```
Fahrkartenschalter Öffnungszeiten: $IFZ Uhr
```

The related entry in file BHFATTRD (german) contains:

```
FS 300 10 Fahrkartenschalter Öffnungszeiten: $IFZ Uhr
```

The link of the attributes is given in file BHFART as follows:

```
8002549 A FS 000000
```

The bitfield no. is 000000 for daily, which can be omitted in **A* and **I*-lines.

After all the definition for the final texts must be added as well.

Each substitute may operate on any number of infotexts with can optionally comprise a validity. In our example *Hafas* should display the opening hours for working days as well as for the weekend.

The related lines in file BHFART are:

```
8000064 I FZ 0000150 000001
8000064 I FZ 0000151 000002
```

The final texts for display are listed in the file INFOTEXT. For our example the lines must read:

```
0000150 7.00 - 19.00
0000151 9.00 - 13.00
```

After a successful transform process *Hafas* will display the following combined text for the stop Celle (german language):

Fahrkartenschalter Öffnungszeiten: 7.00 - 19.00 Uhr (Mo - Fr)

Fahrkartenschalter Öffnungszeiten: 9.00 - 13.00 Uhr (Sa, So)

A corresponding line in file BHFATTRE in the english version would be:

```
FS 300 10 Ticket office hours: $IFZ h
```

so *Hafas* can display with the english language selected the following line:
Ticket office hours: 7.00 - 19.00 h (Mo - Fr)
Ticket office hours: 9.00 - 13.00 h (Sa, So)

8.1.4. Example: multi-storey car park

Combination of an attribue frame text with information texts of different validity.

The attribute frame text for the attribute car park PH is:

```
Parkhaus: $IPS Stellplätze, $IPP pro Tag
```

The related line in file BHFATTR reads:

```
PH 300 10 Parkhaus: $IPS Stellplätze, $IPP pro Tag
```

The link between the attributes is given in file BHFART as follows.

```
8002549 A PH 000000
```

The bitfield no. is 000000 for daily, which can be omitted in **A* and **I*-lines.

In our example *Hafas* should display not only a different price for each day of the week but also a restricted number of places in the period of 01/04/2002 and 14/04/2002 due to construction works.

The lines necessary read in file BHFART as follows:

```
8002549 I PS 0000011 000011
8002549 I PS 0000012 000012
8002549 I PP 0000013 000101
8002549 I PP 0000014 000102
```

The related texts are listed in file INFOTEXT:

```
0000011 120
0000011 120
0000012 100
0000013 10,00 Euro
0000014 15,00 Euro
```

In the end the information given in *Hafas* for the stop Hamburg Hbf may be as follows:

Parkhaus: 120 Stellplätze, 10,00 Euro pro Tag (Mo - Sa; nicht 01.04.2002 - 14.04.2002)

Parkhaus: 120 Stellplätze, 15,00 Euro pro Tag (So; nicht 01.04.2002 - 14.04.2002)

Parkhaus: 100 Stellplätze, 10,00 Euro pro Tag (Mo - Sa; 01.04.2002 - 14.04.2002)

Parkhaus: 100 Stellplätze, 15,00 Euro pro Tag (So; 01.04.2002 - 14.04.2002)

8.1.5. Possibility for multilingual versions of Infotexts

Under normal circumstances the information like opening hours and telephone numbers should be displayed in one language. Nevertheless it's possible to handle multilingual versions.

The basic setting necessary is a different encoding for attribute frame text in different languages in additional files.

Attribute frame text in German listed in file BHFATTRD:

```
HS 300 10 Spezieller Hinweis: $IID
```

Attribute frame text in English listed in file BHFATTRE:

```
HS 300 10 Special Note: $IIE
```

The related lines in file BHFART read:

```
8001421 A HS 000000
```

and for the link to the infotexts:

```
8001421 I ID 0000511 000000
8001421 I IE 0000512 000000
```

The bitfield no. is 000000 for daily, which can be omitted in **A* and **I*-lines.

The text information is stored in file INFOTEXT:

```
0000511 Nach Dettelbach 6km: Weiter mit Bus
0000512 To Dettelbach 6km: Continue by bus
```

After a successful creation of plandata *Hafas* can display the following lines:

German language selected for display:

Spezieller Hinweis: Nach Dettelbach 6km: Weiter mit Bus

English language selected for display:

Special Note: To Dettelbach 6km: Continue by bus

8.1.6. Attributes controlling additional functions of *Hafas*

To enable *Hafas* to calculate on additional information it's possible to assign special attributes.

Example: Diesel operation DF

Stops on non-electrified lines should obtain this additional attribute to show the amout of electric/non-electric operation.

In the file BHFATTR the following line must be present:

```
DF 300 10 Haltestelle liegt an einer nicht elektrifizierten Strecke
```

To apply this attribute to the stops of Heide(Holst), Husum, Westerland(Sylt), Suhl and Cloppenburg the corresponding line in file BHFART should read:

```
8000155 A DF 000000
8000181 A DF 000000
8006369 A DF 000000
8010345 A DF 000000
8001337 A DF 000000
```

The bitfield no. is 000000 for daily, which can be omitted in **A* and **I*-lines.

Example: Online ticketing OT The task is to advertise those stops which enable online ticketing. The file BHFATTR must contain the following entry:

```
OT 300 10 Onlineticket möglich
```

To apply the information to the stops of Bonn, Hannover, Koblenz, Stuttgart und München the file BHFART has to contain the following:

```
8000044 A OT 000000
8000152 A OT 000000
8000206 A OT 000000
8000096 A OT 000000
8000261 A OT 000000
```

The bitfield no. is 000000 for daily, which can be omitted in **A* and **I*-lines.

8.1.7. Definition of the eXtended Infotext XML Format

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- eXtended Infotext XML format -->
<!-- -->
<!-- Applies to all infotexts with code XI -->
<!-- Version history:  ver. 1.0  initial version  -  Stephan Sünderkamp 26.04.07 -->
<!-- ====================================================================== -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:complexType name="XInfoTextParamType">
    <xs:attribute name="n" type="xs:string" use="required">
      <xs:annotation>
        <xs:documentation>Name of the parameter</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="v" type="xs:string" use="required">
      <xs:annotation>
        <xs:documentation>Value of the parameter</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="XInfoTextElementType">
    <xs:sequence>
      <xs:element name="P" type="XInfoTextParamType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            This element represents a parameter, an arbitrary piece of information,
            which is described by a name and a value.
            It is recommended to define standard names for certain contents to
            facilitate automatic processing of the information.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="attributeGroup.XI"/>
  </xs:complexType>
  <xs:element name="XI" type="XInfoTextElementType">
    <xs:annotation>
      <xs:documentation>
        This element groups different parameters of the same type.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:attributeGroup name="attributeGroup.XI">
    <xs:attribute name="t" type="xs:string" use="required">
      <xs:annotation>
        <xs:documentation>Describes type of information</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:attributeGroup>
</xs:schema>
```