



Eine Dokumentation im Umgang mit Traffic

Traffic ist ein programmierbarer Bildschirmschoner, der es ermöglicht eigene Fahrpläne zu erstellen, und eigene Fahrzeuge in das Programm zu integrieren. Die Funktionen und Gestaltungsmöglichkeiten sind so umfangreich, dass es dann erforderlich wird, sich mit den grundlegenden Eigenschaften des Programms vertraut zu machen.

Wer Traffic zunächst mit dem Traffic-Fahrplan nutzen möchte und das Programm zum ersten Mal im Konfigurationsmodus öffnet, muss nicht viele Einstellungen im Konfiguration-Fenster vornehmen.

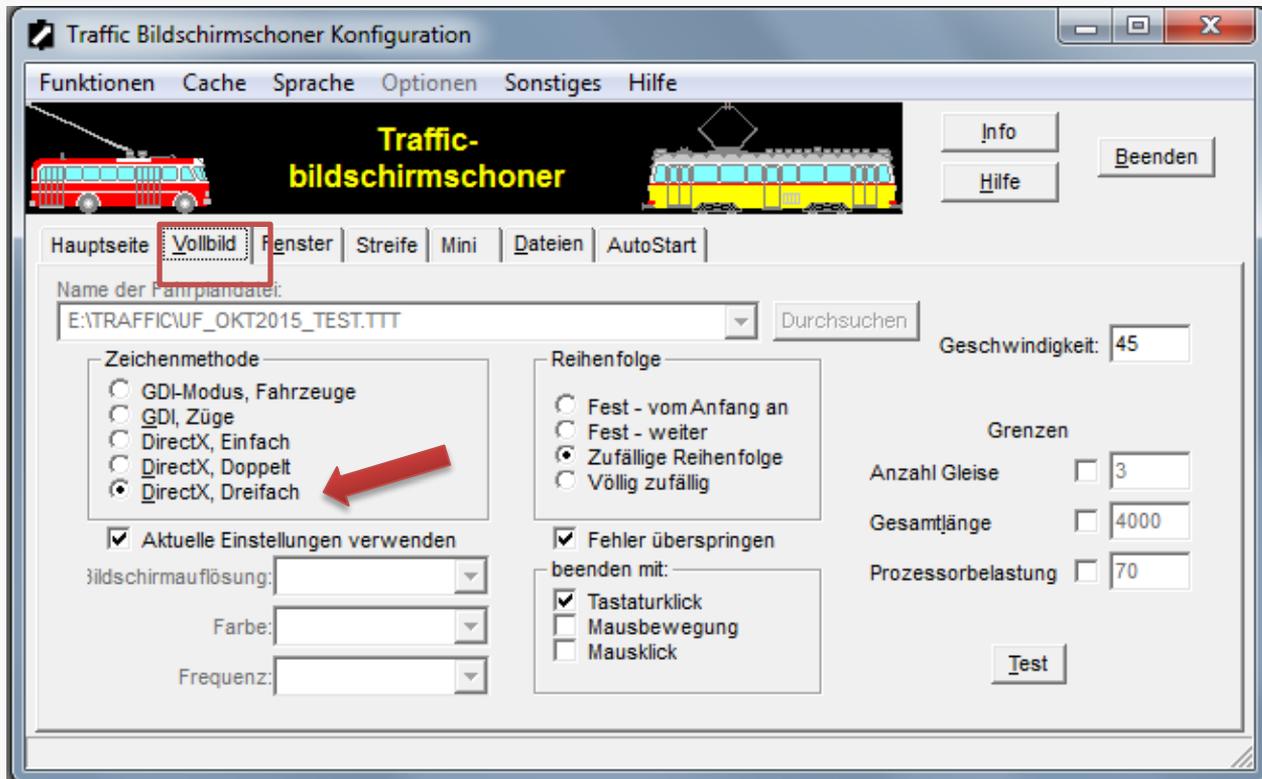
In der zweiten Registerkarte |Vollbild| findet man die wesentlichen Einstellungen zur Bildschirmauflösung und der Zeichenmethode. Der Vollbildmodus verträgt bei neueren Computern durchaus eine DirectX, dreifach Pufferung. Dann sind die Bewegungen sehr flüssig und ruckeln nicht. Die Bildschirmauflösung richtet sich natürlich auch nach den Eigenschaften verwendeten der Grafikkarte und des Monitors. Hier kann man Testen, welche Einstellungen Gefallen finden. Ansonsten kann man die aktuellen Einstellungen ankreuzen, dann nutzt Traffic die augenblicklich eingestellte Bildschirmauflösung.

Traffic ist aber Weit aus mehr als nur ein Bildschirmschonerprogramm. Es ist ein Projekt, das vom Programmautor **Zoltán Szabó** stets aktualisiert wird. Dabei werden die Funktionen des Programms sowie die enorme Fahrzeugbildersammlung, die ein Hauptbestandteil von Traffic ist, fortwährend erweitert.

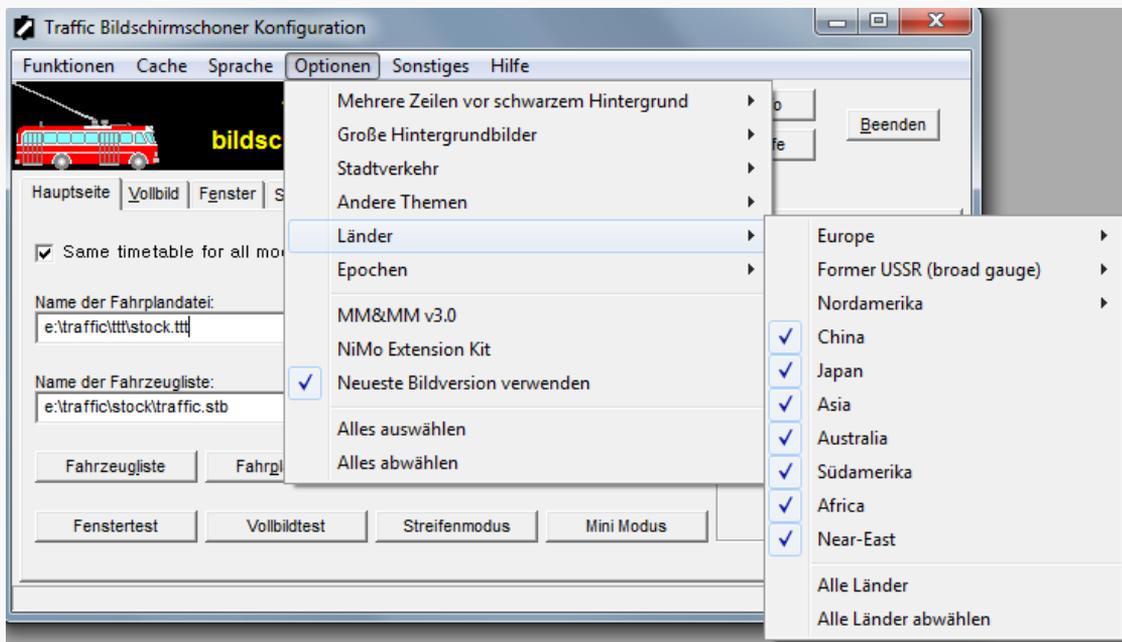
Diese Fahrzeugbildersammlung ist wie Bibliotheks-Katalog strukturiert, und sicher einzigartig in dieser Größe und Vielfalt. Das Programm selbst, und die Fahrzeugbildersammlung sind im Internet frei verfügbar. Unter http://traffic.pxtr.de/zip/traffic_dnld.exe wird ein Installer herunter geladen, mit dem man eine Internet basierte Installation vornehmen kann. Die Bild-Autoren der Fahrzeugbildersammlung schützen sich durch Copyright. Daraus ergibt sich, dass Traffic für eine kommerzielle Nutzung nicht vorgesehen ist.



Das Konfigurations-Fenster ist im Prinzip selbsterklärend. Für die Hilfe-Datei benötigt man allerdings das Windows-Hilfeprogramm ("WinHlp32.exe"), das auf der [Microsoft-Internetseite](#) kostenlos zur Verfügung steht.



Unter Optionen hat man die Möglichkeit den Ablauf des Programms durch Auswahl von Ländern und Epochen zu beeinflussen.



Mehr muss nicht eingestellt werden um Traffic als Bildschirmschoner zu nutzen.

Wer in Traffic einen eigenen Fahrplan erstellen möchte, um einen individuellen Ablauf zu erreichen, benötigt die Fahrzeugliste, den Fahrplan-Editor und den Gliederungsbaum als zusätzliche Fenster.

Die gelben Felder sind Drag&Drop Felder mit unterschiedlicher Funktion.

Es entsteht eine neue Zeile. Schließt man diese mit einem Semikolon statt mit Komma, dann hat man bereits eine funktionierende Fahrplanzeile.

Die Fenster sind frei beweglich und können z.B. wie oben abgebildet angeordnet sein. Traffic merkt sich die Anordnung für den nächsten Aufruf der Konfiguration.

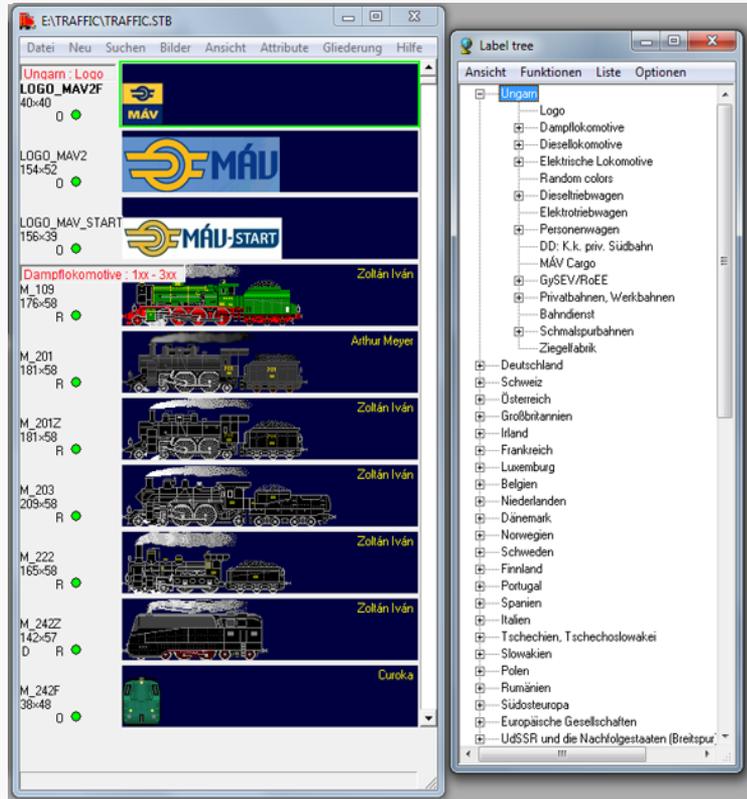
Der Gliederungsbaum lässt sich mit **F11** aufrufen. Einen neuen Fahrplan erhält man durch Klicken auf **Funktionen → Datei → Neue Fahrplandatei**. Die Vorgehensweise zum Schreiben eines eigenen Fahrplans wird im Fahrplan Tutorium beschrieben. Den Traffic-Fahrplan (stock.ttt) sollte man nicht bearbeiten, denn er ist Teil einer komplexen Fahrplan- Struktur.

Unter Optionen im Gliederungsbaum findet man „Mausklick bewegt Fahrzeugliste“, da bitte ein Häkchen dran. Dadurch ist eine bessere Navigation möglich. Es lohnt sich, sich mal ein bisschen durch die Fahrzeugliste zu klicken, um den Umfang der Liste zu erkunden. Der Aufbau ist logisch und differenziert, und es gibt viel zu entdecken, was die Eisenbahnen dieser Welt betrifft.

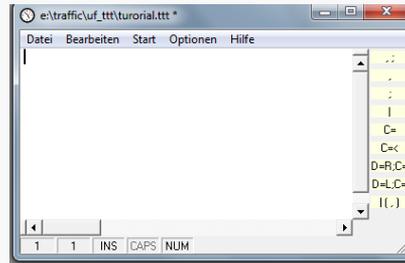
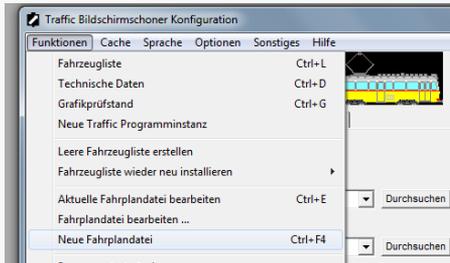
Die Fahrzeugliste ist aus mehreren Bibliothek-Dateien zusammengesetzt. Es ist aber auch möglich, sie mit eigenen Bildern zu ergänzen. Dabei ist zu beachten, dass diese Bilder nicht im Traffic-Ordner gespeichert werden sollten, damit sie bei einer möglichen Deinstallation von Traffic nicht verloren gehen.

Die günstigsten Dateiformate für eigene Bilder sind *.gif, *.bmp (8bit) und *.png, damit die Transparenz des Hintergrunds gewährleistet ist. Um das zu sehen, kann man die Hintergrundfarbe der Fahrzeugliste einstellen. Das Einfügen von eigenen Bildern wird im Kapitel **Modifikationen** nochmal ausführlicher erklärt.

Großformate von Hintergrundbildern müssen nicht in die Fahrzeugliste eingefügt werden. Eine Pfadangabe im Fahrplan reicht aus, um diese Bilder darzustellen, dadurch sind auch andere Formate wie z.B. *.jpg möglich.



Eine wesentliche Eigenschaft von Traffic ist das Erstellen und Bearbeiten eigener Fahrpläne, die auch miteinander verbunden werden können. So kann eine stetig wachsende Struktur aus vielen Fahrzeug-Bewegungen und Fahrplänen entstehen. Man erstellt eine neue Fahrplandatei, indem man im Konfigurationsfenster auf Funktion -> Neue Fahrplandatei klickt. Das neu entstandene Fenster ist eine leere noch nicht gespeicherte Fahrplan-Datei im Fahrplanelitor.



Dieser Editor ist im Prinzip ein Texteditor mit zusätzlichen Funktionen am rechten Rand. Diese gelb unterlegten Flächen sind Drop-Felder, auf die man mit gedrückter Maustaste Bilder aus der Fahrzeugliste ziehen kann. Dadurch entsteht eine neue Zeile mit der entsprechenden Funktion.

Da ein schnelles Ergebnis erzielt werden soll, werden einige Funktionen an dieser Stelle möglicherweise nur teilweise erklärt. Ein Fahrplan besteht im Prinzip aus Bewegungszeilen. Um eine Bewegung zu erzeugen reicht es aus, eine Zeile mit C=<Fahrzeug> zu beginnen und mit einem Semikolon zu schließen. Auf dem Bildschirm sieht man dann dieses Fahrzeug von links nach rechts und von rechts nach links fahren. Sind es mehrere Fahrzeuge (also ein Zug), dann werden diese in der Reihenfolge gezeigt, wie sie im Fahrplan geschrieben wurden. Bei einer Fahrt von rechts nach links wird die Reihenfolge vertauscht, damit die Lok auch hier vorn zu sehen ist. Bei einer E- oder Diesel-Lok mit Wagen sieht das im Prinzip gleich aus, bei einer Dampflok hingegen, fährt die Lok zwar



voraus, aber anders herum. Bei dieser BR 74 kann man das so akzeptieren, schließlich wird das ja in der Realität so gemacht. Ein [<R] vor der Lok ändert das. Der Zug wird dann gespiegelt dargestellt und die Reihenfolge der Phasenbilder umgekehrt.

Bei einem Trieb-Zug hingegen, sieht das generell sehr merkwürdig aus. Ein Einfaches < Zeichen vor dem Zug reicht hier aus, um die



Reihenfolge bei einer Fahrt von rechts nach links **nicht** zu vertauschen.

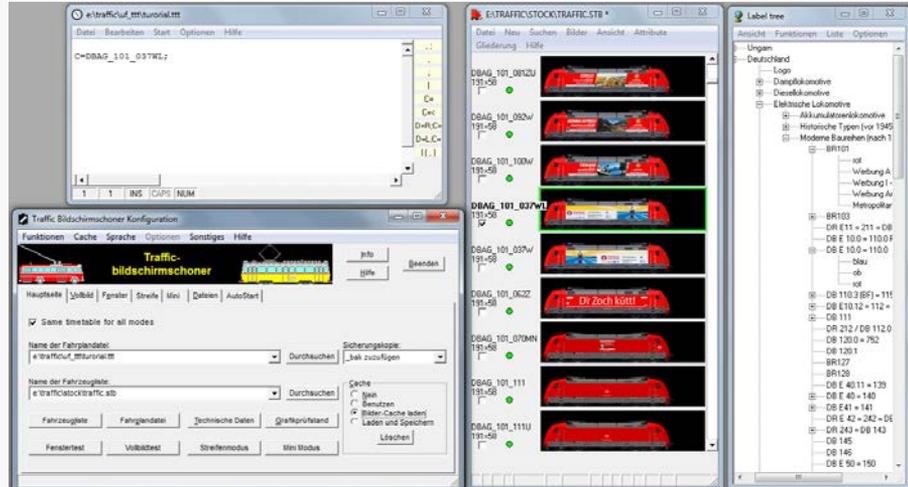
Beginnen wir also mit den ersten eigenen Fahrplanzeilen. Die dazu benötigten Fenster, können in etwa wie unten abgebildet angeordnet sein. Die Fahrzeugliste, die neue Fahrplandatei und der Gliederungsbaum, den man mit F11 aufrufen kann.

Zur besseren Navigation sollte „*Maus bewegt Fahrzeugliste*“ unter Option angehackt sein.

Ziel der ersten Fahrplan-Zeilen ist es, einen IC-Zug mit einer individuellen Trasse fahren zu lassen. Nicht immer den gleichen, sondern eine zufällige von Traffic bestimme Auswahl.

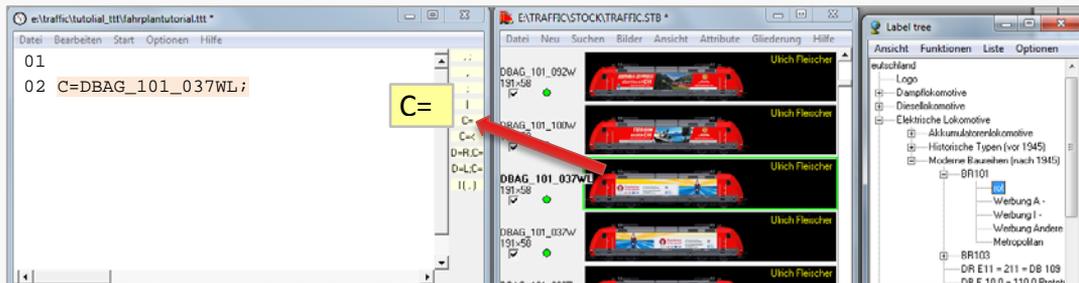
Dazu klickt man im Gliederungsbaum auf Deutschland -> Elektrische Lokomotiv -> Moderne Baureihen (nach 1945) -> BR101 -> rot.

In der Fahrzeugliste ein wenig nach unten scrollen, bis die gewünschte Lok zu sehen ist.



Nun zieht man mit gedrückter Maustaste die markierte Lok auf das Dropfeld **C=** und lässt die Maustaste los.

Es entsteht eine neue Fahrplanzeile in der **C=DBAG_101_037WL;** steht. Das ist bereits eine funktionierende Fahrplanzeile, ohne dass man selbst etwas schreiben musste.

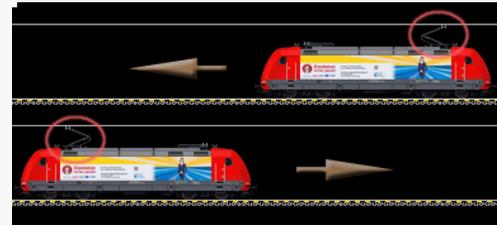


Nun wird die Datei in Traffic in einem Ordner (z.B. Eigene_ttt) mit einem Namen ohne Umlaute und Leerzeichen gespeichert. Zum Testen im Vollbildmodus muss dieser Fahrplan nun noch im Konfigurationsfenster eingetragen werden. Zum Testen im Fenstermodus reicht die **F5** Taste.



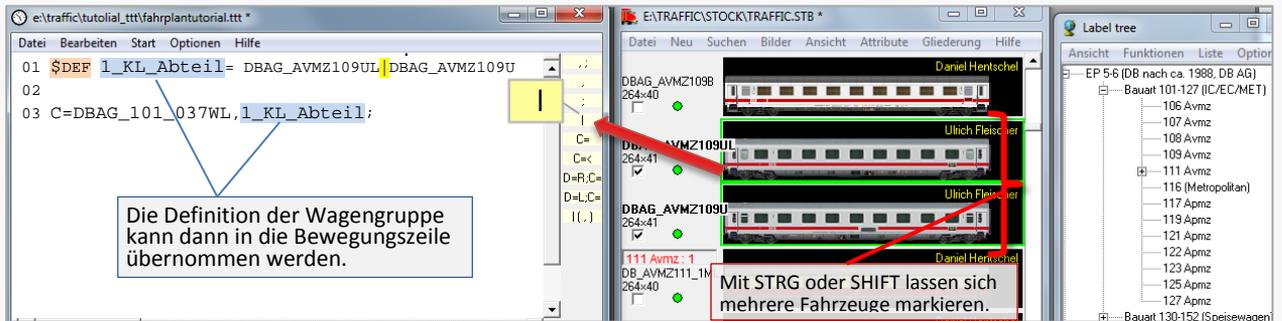
Das Ergebnis entspricht noch nicht dem Ziel, aber es fällt auf, dass die Stromabnehmer unabhängig von der Fahrtrichtung immer die richtige Stellung haben.

Es fehlen noch die Wagen und eine individuelle Trasse. Aber es ist ein Anfang und die Zeile kann nun ergänzt werden. Dazu navigiert man in der Gliederung nach Deutschland ->



Personenwagen -> EP 5-6 -> Bauart 101-127 -> 108Avmz. In der Fahrzeugliste wieder ein bisschen scrollen, bis man die neueren Bilder DBAG_AVMZ108_1UL von mir erreicht.

Um nun nicht jeden IC einzeln zu schreiben, werden die Wagen als Gruppen im Fahrplan definiert. Dazu wird ein **\$DEF <Name>=** vor die Wagengruppen geschrieben. Diese Arbeit macht man natürlich nur einmal, dann sind die IC definiert, und man kann die



Züge im Fahrplan zusammenstellen. Um auch diese Arbeit zu erleichtern, gibt es ein Dropdown mit einem senkrechten Strich | , der die darauf gezogenen Fahrzeuge zu einem gleichberechtigten „oder“ in einer Zeile zusammenstellt. Wichtig bei dieser Arbeit ist, dass die Schreibmarke immer **hinter** dem letzten Zeichen steht, sonst entsteht eine neue Zeile. Diese Zeile kann je nach Anzahl der Fahrzeuge sehr lang werden.

Zur besseren Übersicht, kann man die Zeile mit Hilfe eines Backslash Zeichens \ hinterher aufteilen. Dabei ist darauf zu achten, dass keine weiteren Zeichen hinter dem \ stehen. Auch keine TABS oder Leerzeichen.

Was hierbei auffällt, dass die Navigation recht zeitaufwendig ist, um diesen IC Zug zusammenzustellen. Das liegt natürlich in erster Linie an der großen Menge an Bildern, aber auch an den vielen Wagengruppen des IC. Das ist nicht bei jedem Zug so, und mit dem Kennenlernen der Liste wird das schnell besser. Die so erstellten Definitionen sind immer wieder zu verwenden und einfach zu ergänzen.

Die typische Wagenreihung eines IC besteht aus **Lok**, **1.Kl. Großraumwagen**, **1.Kl. Abteilwagen**, **Bistrowagen(optional)**, **2.Kl. Wagen**, und **Steuerwagen am Ende (optional)**.

Diese Gruppen wurden nun nach oben aufgeführtem Muster definiert. Damit kann diese Definition später schnell findet, ist es sinnvoll, sie zu bezeichnen. Eine Kommentarzeile beginnt immer mit einer Raute # und kann danach beliebige Zeichenfolgen haben.

Um das Tutorium zu beschleunigen, lassen sich die Zeilen aus den grauen Textfeldern in einen Test-Fahrplan kopieren.

```

### Definition IC ###
$DEF 1_Kl_Abteil= DBAG_AVMZ108_1UL|DBAG_AVMZ108_1U
                 |DBAG_AVMZ109UL|DBAG_AVMZ109U
\
$DEF 1_Kl_Gr_Raum= DBAG_APMZ127_1UL|DBAG_APMZ127_1U
$DEF Bistro= DBAG_ARKIMBZ266UL|DBAG_ARKIMBZ266U
$DEF 2_Kl= DBAG_BIMDZ268DL|DBAG_BIMDZ268D|DBAG_BPMZ294UL\
            |DBAG_BPMZ294U|DBAG_BPMZ186UL|DBAG_BPMZ186U
$DEF Steuerwg_L= DBAG_BIMDZF271UL|DBAG_BPMBDZF296L
$DEF Steuerwg_R= DBAG_BIMDZF271U|DBAG_BPMBDZF296
$DEF Lok= DBAG_101_081ZU|DBAG_101_100W|DBAG_101_037W\
           |DBAG_101_062Z|10:DBAG_6_120_101|10:DBAG_101_111
### Ende Definition IC ###
    
```

Was nun noch fehlt, ist die eigentliche Bewegungszeile im Fahrplan. Der erste IC soll keinen Steuerwagen haben und fängt deshalb nur mit einem C= an. Dieser Zug sieht links und rechts fahrend fast identisch aus, ohne zusätzliche Parameter.

C=**Lok**,(0|1)***1_Kl_Gr_Raum**,(1-2)***1_Kl_Abteil**,(0|1)***Bistro**,(3-5)***2_Kl**; **V=8-12**;

Der nächste Zug fängt mit einem Steuerwagen an, deshalb mit C=<, damit die Reihung bei Links- und Rechtsfahrt gleich bleibt.

C=<**Steuerwg_L**,(3-5)***2_Kl**,(0|1)***Bistro**,(1-2)***1_Kl_Abteil**,(0|1)***1_Kl_Gr_Raum**,**Lok**; **V=8-12**;

Beim letzten Zug wird die Lok an den Anfang gesetzt, danach die übliche Reihung mit einem Steuerwagen am Ende. Also wieder

C=<**Lok**,(0|1)***1_Kl_Gr_Raum**,(1-2)***1_Kl_Abteil**,(0|1)***Bistro**,(3-5)***2_Kl**,**Steuerwg_R**; **V=8-12**;

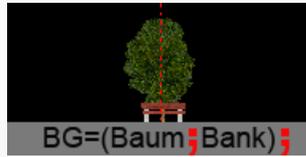
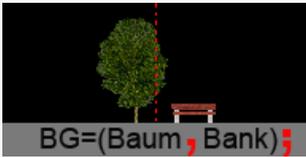
Mit diesen drei Bewegungszeilen hat man viele typische IC abgedeckt, die in Deutschland unterwegs sind.

Jede Bewegungszeile kann auch zusätzliche Parameter besitzen. Ein V=<Wert>; regelt die Geschwindigkeit einer Bewegungszeile.

Damit diese IC-Züge mit vergleichbarem Wert über den Bildschirm fahren, fügt man hinter das Semikolon **V=8-12**; ein.



Um das Ziel der ersten eigenen Fahrplanzeilen abzuschließen, fehlt noch die Trasse. Da hat Traffic jede Menge zu bieten. Die Gestaltung kann im Hinter- und Vordergrund vorgenommen werden und besteht aus Bildern oder Animationen, die ebenfalls in der Fahrzeugliste zu finden sind. Die Gestaltungsmöglichkeiten sind sehr komplex und brauchen manchmal eine ganze Reihe von Parametern. Die grundlegenden Parameter betreffen die Abmessungen der Bewegungszeile. Ohne Angaben hierfür hat eine Zeile die Höhe von 58 Pixeln. Das reicht für die Darstellung einer Oberleitung mit Masten nicht aus. Da man grundsätzlich eine Trasse darstellen möchte, können diese Abmessungsparameter in den Kopf des Fahrplans geschrieben werden. Dazu schreibt man \$ABOVE 30 und \$BELOW 10 jeweils in eine Zeile. Damit hat man 98 Pixel für die Darstellung der Trasse. Außerdem soll die vordefinierte Trasse unsichtbar sein, dazu kann man ebenfalls im Kopf des Fahrplans \$WAY E;R=X schreiben.

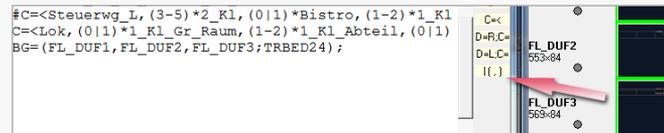


setzt man diese in Klammern. Wie bei den beiden Bildern oben zu sehen ist, gibt es einen großen Unterschied zwischen Komma und Semikolon in der Darstellung. Während das **Komma** die Bilder **reih**t, setzt das **Semikolon** das zweite Bild in eine **neue** darüber liegende **Ebene**. Genau das braucht man für die Darstellung einer Oberleitung im Hintergrund, und einem Gleis davor.

Bisher gibt es drei Bewegungszeilen. Da der Hintergrund Bestandteil jeder einzelnen Bewegungszeile ist, ist es sinnvoll zunächst nur eine Zeile zu testen. Dazu kommentiert man die beiden ersten mit einer Raute # am Beginn der Zeile aus. Hinter die dritte Zeile

schreibt man ein Backslash \, dann mit Return in die nächste Zeile und ein BG= (Background). Im Verzeichnisbaum navigiert man nach Hintergrund -> Oberleitung -> Deutschland und scrollt zu FL_DUF1, FL_DUF2 und FL_DUF3. Die drei markieren und über das letzte Dropfeld ziehen. Dann stehen die Bilder in Klammern durch Komma getrennt. Die Schreibmarke in die Klammer setzen und das Gleis TRBED24 über das Semikolon ziehen. Das Ganze mit einem Semikolon abschließen und der Hintergrund ist definiert. Es fehlen aber noch die Angaben über die Art der Darstellung und der Positionierung. Generell gilt, dass zuerst der horizontale Wert, dann der vertikale Wert bestimmt wird. Die Horizontale braucht in diesem Fall keine Position, da Oberleitung und Gleis bildschirmfüllend dargestellt werden sollen.

Im Gegensatz zu den fahrenden Zügen, haben die Bilder im Hinter- und Vordergrund eine Position. Dabei ist es möglich die Bilder **nebeneinander** als auch **übereinander** darzustellen. Das zuerst Geschriebene wird auch hier zuerst gezeichnet. Macht man keine Angabe zur Position, werden die Bilder in der Mitte des Bildschirms zentriert. Sind es mehrere Bilder,



Dazu gibt es einige Zeichen die nur im Bereich BG= und FG= gelten, oder in einem anderen Zusammenhang eine andere Bedeutung haben. Die korrekte Zeile muss dann so aussehen:

..	Wiederholt Bildschirmfüllend
#	Verschiebt Bilder in einem Rahmen
^0	Oberhalb des Bildes

BG=(FL_DUF1,FL_DUF2,FL_DUF3:..., -3,#;TRBED24:...,^0);

Oberhalb des Gleises fährt der Zug

Hinter dem : beginnt die x,y Ausrichtung

3 Pixel tiefer

Verschiebt die Bilder

Zwei Punkte hintereinander, wiederholt Bildschirm füllend

Damit ist die Trasse zunächst fertig und gilt für die **letzte** Fahrplanzeile. Der gesamte Fahrplan hat jetzt drei Bereiche: Den **Kopf** mit den allgemeinen Fahrplanbefehlen, die für alle Bewegungszeilen gelten; Die eignen **Definitionen** \$DEF=, die sich auf einzelne Fahrplanzeilen beziehen und die eigentlichen Bewegungszeilen C=;BG=;

Eine Definition für den **Hintergrund**, damit dieser für alle Fahrplanzeilen gilt, wäre hilfreich. Und so einen Parameter gibt es! Mit Hilfe des „Zauber-Parameters“ **AP=** (Additional Parameter) kann man am Ende jeder Fahrplanzeile eine Definition aufrufen in der nicht nur Hinter- und Vordergrundeigenschaften stehen, sondern auch weitere Trassenparameter und sogar einige Bewegungsarten enthalten sein können.

Dazu gibt man dem gesamtem Hintergrund **BG=**, ein sinnvolles Kürzel wie z.B. **TR1F** (Trasse1Fahrdraht) und setzt ein \$DEF davor.

\$DEF **TR1F**=[BG=(FL_DUF1,FL_DUF2,FL_DUF3:..., -3,#;TRBED24:...,^0);] [V=9-11]

Die gesamte Hintergrunddefinition wird dazu lediglich in eckige Klammern gesetzt.

Statt des langen Hintergrundbegriffs steht jetzt am Ende jeder Bewegungszeile ein **AP=TR1F**.

Man kann nun \$DEF **TR1F** ergänzen oder verändern, ohne die Änderung in jede einzelne Bewegungszeile zu schreiben.

Das wird zunächst mit der Geschwindigkeitsangabe gemacht, denn die gilt auch für jede einzelne Bewegungszeile. [V=9-11].

Auch hier wird der Ausdruck in eckige Klammern gesetzt.

```

$ABOVE 30
$BELOW 10
$SWAY E;R=X
### Definition IC ###
$DEF 1_Kl_Abteil= DBAG_AVMZ108_1UL|DBAG_AVMZ108_1UL\
|DBAG_AVMZ109UL|DBAG_AVMZ109U
$DEF 1_Kl_Gr_Raum= DBAG_APMZ127_1UL|DBAG_APMZ127_1U
$DEF Bistro= DBAG_ARKIMB2266UL|DBAG_ARKIMB2266U
$DEF 2_Kl= DBAG_BIMDZ2268DL|DBAG_BIMDZ2268D|DBAG_BPMZ294UL\
|DBAG_BPMZ294U|DBAG_BPMZ186UL|DBAG_BPMZ186U
$DEF Steuerwg_L= DBAG_BIMDZ271UL|DBAG_BPMBDZF296L
$DEF Steuerwg_R= DBAG_BIMDZ271U|DBAG_BPMBDZF296
$DEF Lok= DBAG_101_081ZU|DBAG_101_100W|DBAG_101_037W\
|DBAG_101_062Z|10:DBAG_6_120_101|20:DBAG_101_111
### Ende Definition IC ###
C=Lok,1_Kl_Gr_Raum,1_Kl_Abteil,(0|1)*Bistro,(3-5)*2_Kl;V=8-12;
C=<Steuerwg_L,(3-5)*2_Kl,(0|1)*Bistro,(1-2)*1_Kl_Abteil,(0|1)*1_Kl_Gr_Raum,Lok;V=8-12;
C=<Lok,(0|1)*1_Kl_Gr_Raum,(1-2)*1_Kl_Abteil,(0|1)*Bistro,(3-5)*2_Kl,Steuerwg_R;V=8-12;]
BG=(FL_DUF1,FL_DUF2,FL_DUF3:..., -3,#;TRBED24:...,^0);
    
```

So eine Trassendefinition eignet sich gut, um die Bewegung **M=STOP**; aufzunehmen, da diese Bewegung auch ohne Angabe zusätzlicher Parameter gut funktioniert. Die unterschiedlichen Fahrzeug-Bewegungen werden im Kapitel **Bewegungen** ausführlich mit Beispielen beschrieben. Zum Probieren wird folgende neue Definition geschrieben:

`§DEF TR1FS=[BG=(FL_DUF1,FL_DUF2,FL_DUF3:..., -3, #; TRBED24:..., ^0);] [V=9-11] [M=M3]`

Hier zeigt sich, das ein gutes Kürzel hilfreich sein kann. Der ursprüngliche *Name* der Trassendefinition **TR1F**, bekommt nur ein **S** für **STOP** als Namensergänzung.

Nun sind nicht alle Bewegungsarten für alle Züge gleichermaßen geeignet, dazu sind die Bewegungen selbst, aber auch die Züge zu spezifisch. **M=STOP**; funktioniert aber bei jeder Art von Fahrzeug, und das auch unabhängig von der Fahrtrichtung. Wie man im Fahrplan (rechts) erkennen kann, lassen sich Definitionen auch in Kombination aufrufen.

`AP=5:TR1F|1:TR1FS`

Um Die Häufigkeit der Trassen zu steuern, wurden Zahlen gefolgt von einem Doppelpunkt eingesetzt. Das wurde auch bei `Lok=` so gemacht, um das Verhältnis zwischen Werbeloks den roten 101 und der einen 120 zu verändern.

Auch die globalen Fahrplanbefehle wurden um zwei weitere Befehle erweitert. `§IV 0.5` sorgt für weiches Anfahren und `§BGCOLOR #000000` hält den Bildschirm schwarz, auch wenn in der Fahrzeugliste eine andere Hintergrundfarbe eingestellt ist.

Die jetzige Struktur des Fahrplans geeignet sich gut, um weitere Züge in den Fahrplan zu schreiben. Es ist sozusagen ein „Gerüst“, das man sich in seinen eigenen Fahrplan kopieren kann. Man hat eine beschreibende Trassen-Definition, die für alle Bewegungszeilen gilt, und dadurch sind die eigentlichen Bewegungszeilen recht kurz.

```

§BGCOLOR #000000
§IV 0.5
§ABOVE 30
§BELOW 10
$WAY E;R=X
### Definition IC ###
$DEF 1_Kl_Abteil= DBAG_AVMZ108_1UL|DBAG_AVMZ108_1U|\
DBAG_AVMZ109UL|DBAG_AVMZ109U
$DEF 1_Kl_Gr_Raum= DBAG_APMZ127_1UL|DBAG_APMZ127_1U
$DEF Bistro= DBAG_ARKIMBZ266UL|DBAG_ARKIMBZ266U
$DEF 2_Kl= DBAG_BIMDZ268DL|DBAG_BIMDZ268D|DBAG_BPMZ294UL|\
DBAG_BPMZ294U|DBAG_BPMZ186UL|DBAG_BPMZ186U
$DEF Steuerwg_L= DBAG_BIMDZF271UL|DBAG_BPMBDZF296L
$DEF Steuerwg_R= DBAG_BIMDZF271U|DBAG_BPMBDZF296
$DEF Lok= DBAG_101_081ZU|DBAG_101_100W|DBAG_101_037W|\
DBAG_101_062Z|10:DBAG_6_120_101|10:DBAG_101_111
### Ende Definition IC ###

### Definition Trassen ###
$DEF TR1F=[BG=(FL_DUF1,FL_DUF2,FL_DUF3:..., -3, #; TRBED24:..., ^0);][V=8-12]
$DEF TR1FS=[BG=(FL_DUF1,FL_DUF2,FL_DUF3:..., -3, #; TRBED24:..., ^0);][V=8-12]\
[M=M3]
### Ende Definition Trasse ###

C=Lok,1_Kl_Gr_Raum,1_Kl_Abteil,(0|1)*Bistro,(3-5)*2_Kl;AP=10:TR1F|1:TR1FS
C=<Steuerwg_L,(3-5)*2_Kl,(0|1)*Bistro,(1-2)*1_Kl_Abteil,(0|1)*1_Kl_Gr_Raum,Lok;\
AP=10:TR1F|1:TR1FS
C=<Lok,(0|1)*1_Kl_Gr_Raum,(1-2)*1_Kl_Abteil,(0|1)*Bistro,(3-5)*2_Kl,Steuerwg_R;\
AP=10:TR1F|1:TR1FS
    
```

Nun ist festzustellen, dass das Zusammenstellen von Zügen und die Definition in Wagengruppen eine recht aufwendige Angelegenheit sein kann. Das hängt vor allem von der Art des Zugs ab. Es ist nicht immer so komplex wie bei dem Beispiel IC. Triebzüge z.B. fahren meist in festen Reihungen, da ist der Aufwand geringer. Stehen die Züge aber erst einmal im Fahrplan, dann ist anhand der Listen-Nr. kaum noch zu erkennen, um welches Fahrzeug es sich handelt.

Dazu gibt es eine sehr hilfreiche Tastenkombination. Im Fahrplan setzt man die Schreibmarke **in** oder **hinter** den Bildnamen, den es zu identifizieren gilt und drückt dann das Plus-Zeichen auf der numerischen Tastatur, und die Fahrzeugliste springt zu dem gesuchten Bild. Wichtig dabei ist, dass es tatsächlich das numerische Plus-Zeichen sein muss, sonst wird ein + in den Fahrplan geschrieben. Bei Notebooks ist das üblicherweise eine Fn-Taste.

Die Definition von Wagengruppen ist aber generell eine sinnvolle Sache. Die Liste der Güterwagen ist so umfangreich, dass man viele solcher Definition erstellen kann, die man dann z.B. in einer externen Fahrplan-Datei „**G_Wag_Ep6.ttt**“ speichert. Eine solche Datei ist dann eine Fahrplan-Datei **ohne** Bewegungszeilen die nur Definitionen enthält. Die Vorgehensweise ist dann:

```

###Schiebewandwagen(Habis, Habils, Habins)###
$DEF HABISS_1 = SBB_2750000|SBB_2750000A|SBB_2750000Y|SBB_2750000Y1|\
SBB_2750000_VOLG|SBB_2750000_VOLG1|SBB_2750500A|\
SBB_2750500GR1|SBB_2750500GR2|SBB_2850500A|\
TWAG_HABIS802753499L|TWAG_HABIS802753499|DB_HABIS6RS1|DB_HABIS6RS2|\
DB_HABIS6RS3|G_009|G_010|G_014
$DEF HABISS_2 = DB_HABIS7TW1|DB_HABIS7TW2|DB_HABIS7TW3|DB_HABIS7TW4|DB_HABIS7TW5|\
DB_HABIS2|TWAG_HABIS2|TWAG_HABIS3L|TWAG_HABIS3|DB_HABIS8RS1|DB_HABIS8RS2|\
DB_HABBIILINS2802004

###Wagen mit öffnungsfähigem Dach(Tapps, Tagpps)###
$DEF TAGNPP = SBB_TAGNPPS009L|SBB_TAGNPPS009|SBB_TAGNPPS010L|SBB_TAGNPPS010|SBB_TAGNPPS011L|\
SBB_TAGNPPS012L|SBB_TAGNPPS012|SBB_TAGNPPS0664025L|SBB_TAGNPPS0664025|\
SBB_TAGNPPS0664038L|SBB_TAGNPPS0664038|SBB_TAGNPPS0664115L|SBB_TAGNPPS0664115|\
SBB_TAGNPPS0664109L|SBB_TAGNPPS0664109|SBB_TAGNPPS0664166L|SBB_TAGNPPS0664166|\
SBB_TAGNPPS0664178L|SBB_TAGNPPS0664178|SBB_TAGNPPS0664198L|SBB_TAGNPPS0664198

```

Möchte man nun einen Güterzug zusammenstellen, dann schreibt man in den Kopf des aktuellen Fahrplans \$REQUIRE **G_Wag_Ep6.ttt**. In der Bewegungszeile des aktuellen Fahrplans braucht man dann nur **TAGNPP**, **HABISS_2**, **HABISS_2** zu schreiben, und es wird eine zufällige Auswahl aus dieser Wagengruppe auf dem Bildschirm zu sehen sein. Für diese Arbeit kann man die Fenster so anordnen, dass man zwei Fahrplan-Dateien zur Ansicht hat, um die Definitionen in den aktuellen Fahrplan zu kopieren.

The screenshot displays three windows from the Traffic software:

- Left Window (e:\traffictutorial\tttig_wag_ep6.ttt):** A text editor showing the definition of vehicle types. Line 02 defines `HABISS_1` as a set of SBB train types. Line 07 defines `HABISS_2` as another set of SBB train types. Line 11 defines a general `Wagen` type with a roof. Line 12 defines `TAGNPF` as a set of specific train types, including `SBB_TAGNPPS0664025L`.
- Middle Window (E:\TRAFFIC\STOCK\TRAFFIC.STB):** A vehicle list window showing a grid of train images. The first row shows `SBB_TAGNPPS012L` (200:44) with a green dot. The second row shows `SBB_TAGNPPS012` (200:44) with a green dot. The third row shows `SBB_TAGNPPS0664025L` (200:44) with a green dot. The fourth row shows `SBB_TAGNPPS0664025` (200:44) with a green dot. The fifth row shows `SBB_TAGNPPS0664038L` (200:44) with a green dot. The sixth row shows `SBB_TAGNPPS0664038` (200:44) with a green dot. The seventh row shows `SBB_TAGNPPS0664115L` (200:44) with a green dot. The eighth row shows `SBB_TAGNPPS0664115` (200:44) with a green dot. The ninth row shows `SBB_TAGNPPS0664109L` (200:44) with a green dot. The status bar at the bottom shows `T R 200:44 155 4492 B`.
- Right Window (Label tree):** A hierarchical tree view showing the structure of the vehicle list. The root is `Nordamerika`, followed by `Wagen`, `Australien und Neuseeland`, `Africa und der Nahe Osten`, `Asien`, `Mittlere- und Südamerika`, `Ladung`, and `Güterwagen`. Under `Güterwagen`, there are sub-entries for `Beladener Wagen`, `Autotransport`, `Offene Wagen in Regelbauart`, `Offene Wagen in Sonderbauart`, `Offene Wagen in Sonderbau`, `Gedeckte Wagen in Regelbauart`, `Gedeckte Wagen in Sonderbauart`, `Gedeckte Wagen in Sonderbau`, `Flachwagen (I)`, `Flachwagen in Regelbauart, 2`, `Flachwagen in Sonderbauart, 2`, `Offene Wagen, die als Flachw.`, `Drehgestellflachwagen in Reg`, `Drehgestellflachwagen in Son`, and `Wagen mit öffnungsfähigem D`. Under `Wagen mit öffnungsfähigem D`, there are sub-entries for `Klappdeckelwagen`, `Schiebe- od. Schwenk`, `Tbis`, `Andere`, `Tds`, and `Tdgs, Tgpps, Tpps`. Under `Wagen mit öffnungsfähigem D`, there are sub-entries for `Bedeckte Trichterwagen d. Sc`, `Silowagen mit Be-/Entladung`, `Flachwagen f. hohe od. schw`, `Sonderwagen (L)`, `Kesselwagen (Z)`, and `Güterunterbedeckthawagen`.

Die Fahrzeugliste dient dann als Katalog, aus dem man sich seinen Wunschzettel zusammenstellt, und als Definitionen in Dateien speichert. Dabei kann man eine individuelle Auswahl treffen je nach Geschmack. Schreibt man dann später in seinen Fahrplan z.B. `7*HABISS_1`, dann wählt Traffic sieben Wagen aus der `HABISS_1` Definition aus.

Der bisherige Beispielfahrplan sieht dann mit erweiterter Trassen-Definition so aus:

Unter den globalen Fahrplanbefehlen steht nun \$REQUIRE G_Wag_Ep6.ttt. Diese Datei enthält ausschließlich Güterwagen Definitionen, die beliebig mit weiteren Definitionen erweitert werden kann.

Unterhalb der bisherigen Fahrplanzeilen wurde eine Güter Lok Definition G_LOK erstellt, damit nicht immer die gleiche Lok vor den Wagen zu sehen ist.

Die letzte Zeile im Beispielfahrplan ist eine Bewegungszeile mit Güterwagen, die aus der Definitions-Datei G_Wag_Ep6.ttt stammen.

Die Trassendefinition wurde mit \$DEF Busch und \$DEF Fahrdraht ergänzt.

Es wird deutlich, dass die Definitionen in dem Beispielfahrplan mehr Platz benötigen, als die eigentlichen Bewegungszeilen, denn das sind nur vier. Es ist also durchaus von Nutzen, wenn man mehrere Fahrplan-Dateien erstellt, die nur Definitionen enthalten. Auf diese Weise hat man dann keine festen Züge, sondern immer eine zufällige Zusammenstellung der Fahrzeuge, was für Abwechslung und Vielfalt sorgt.

```

$BGCOLOR #000000
$IV 0.5
$ABOVE 30
$BELOW 10
$WAY E;R=X
$REQUIRE G_Wag_Ep6.ttt

### Definition IC ###
$DEF 1_Kl_Abteil= DBAG_AVMZ108_1UL|DBAG_AVMZ108_1U|\
DBAG_AVMZ109UL|DBAG_AVMZ109U
$DEF 1_Kl_Gr_Raum= DBAG_APMZ127_1UL|DBAG_APMZ127_1U
$DEF Bistro= DBAG_ARKIMBZ266UL|DBAG_ARKIMBZ266U
$DEF 2_Kl= DBAG_BIMDZ268DL|DBAG_BIMDZ268D|DBAG_BPMZ294UL|\
DBAG_BPMZ294U|DBAG_BPMPZ186UL|DBAG_BPMPZ186U
$DEF Steuerwg_L= DBAG_BIMDZF271UL|DBAG_BPMBDFZ296L
$DEF Steuerwg_R= DBAG_BIMDZF271U|DBAG_BPMBDFZ296
$DEF Lok= DBAG_101_081ZU|DBAG_101_100W|DBAG_101_037W|\
DBAG_101_062Z|10:DBAG_6_120_101|10:DBAG_101_111

### Ende Definition IC ###

### Definition Trassen ###
$DEF Fahrdraht= FL_DUF1,FL_DUF2|FL_DUF3
$DEF Busch= BUSH20|#BUSH20
$DEF TR1F=[BG=( Busch:<10-40+,-8;Busch:<60-80+,-8; Fahrdraht:...;3,#;TRBED24:...;^0);]\
[V=8-12]
$DEF TR1FS=[BG=( Busch:<10-40+,-8;Busch:<60-80+,-8; Fahrdraht:...;3,#;TRBED24:...;^0);]\
[V=8-12] [M=M3]
### Ende Definition Trassen ###

C=Lok,1_Kl_Gr_Raum,1_Kl_Abteil,(0|1)*Bistro,(3-5)*2_Kl;AP=10:TR1F|1:TR1FS
C=<Steuerwg_L,(3-5)*2_Kl,(0|1)*Bistro,(1-2)*1_Kl_Abteil,(0|1)*1_Kl_Gr_Raum,Lok;\
AP=10:TR1F|1:TR1FS
C=<Lok,(0|1)*1_Kl_Gr_Raum,(1-2)*1_Kl_Abteil,(0|1)*Bistro,(3-5)*2_Kl,Steuerwg_R;\
AP=10:TR1F|1:TR1FS

$DEF G_LOK=RPOOL_193801RETRL|RPOOL_193801SPITZKEL|MRCE_185_545R4C|\
CNX_E37_501L|DISP_ES64U2_018

C=(1|2)*G_LOK,8-12*HABISS_2|HABISS_2|TAGNPP;AP= TR1F
    
```

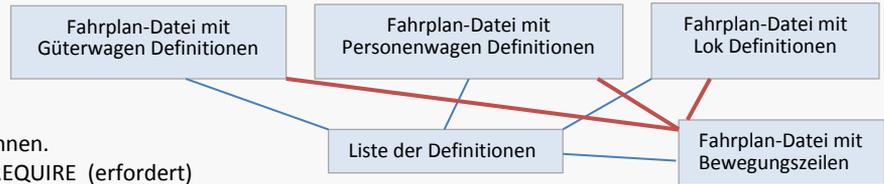
Um nicht den Überblick zu verlieren, kann man zusätzlich eine Liste der Definitionen Erstellen:

Diese Liste kann Definitionen aus mehreren Dateien enthalten, und dient dann als „Stichwortverzeichnis“. Bei wachsender Anzahl von Fahrzeug-Definitionen kann das sehr hilfreich sein.

```

/// $DEF Liste Güterwagen Epoche 5-6 Europa///
HABISS_1 = Schiebewandwagen(Habis, Habils, Habins)
HABISS_2 = Schiebewandwagen(Habis, Habils, Habins)
TAGNPP = Wagen mit öffnungsfähigem Dach(Tapps, Tagpps)
    
```

Die nebenstehende Grafik ist natürlich nur ein Beispiel für eine Struktur aus mehreren Fahrplänen. Es soll lediglich gezeigt werden, dass Fahrpläne in Beziehungen stehen können. Dazu nutzt man die Anweisungen \$REQUIRE (erfordert) und \$INCLUDE (beinhaltet) um Fahrplandateien zu verbinden.



Noch ein paar Worte zur Syntax. Grundsätzlich sollten innerhalb von Ausdrücken und Definitionen Leerzeichen vermieden werden. Meistens funktioniert es dann zwar trotzdem, aber es führt zu Fehlermeldungen in der Fehlerdatei. Ansonsten kann der Fahrplan jede Menge TAB und Leerzeichen zur besseren Lesbarkeit haben, solange diese nicht innerhalb der Ausdrücke oder dahinter stehen, wenn man eine Zeile aus Platzgründen mit \ trennt.

Das Gleiche gilt auch für die Bezeichnungen von Dateien und Ordnern. Lieber ein Unterstrich als ein Leerzeichen. Bei den Namen und Bezeichnungen bitte auch keine Umlaute benutzen, auch das gilt für Ordner innerhalb der Traffic Verzeichnisstruktur. Ein Ordner mit dem Namen Fahrpläne ist also ungünstig - für Traffic sind das dann Fahrpl?ne.

In diesem Sinn viel Vergnügen
Ulrich Fleischer

Das Wesentliche in Traffic sind Bewegungszeilen. Ist keine Bewegungszeile definiert, dann bleibt der Bildschirm schwarz. Diese Bewegungszeilen werden im Grundsatz mit **M=<Bewegung>;C=<Zug>**; erzeugt.

Achtzehn unterschiedliche Bewegungen sind in Traffic definiert. Diese vordefinierten Abläufe der Bewegungen funktionieren auch ohne Angabe zusätzlicher Parameter. Einige der Bewegungen beinhalten allerdings mehrere Züge oder Zugteile, so das man, sollte das der Fall sein, um die Angabe von C=; C1=; C2= und C3=; nicht herumkommt. Zum Ablauf der Bewegungen reicht diese Angabe aber aus.

Die erweiterten Möglichkeiten von Traffic nutzt man aber erst, wenn man mit zusätzlichen Parametern und Modifikationen eine individuelle Trasse gestaltet, die diese unterschiedlichen Bewegungen akzentuiert. Dabei wird ins besondere der T=<Wartezeit>; Parameter interessant, der diese Wartezeit in Bereiche teilt: T=<warten>,<Zeit in Sekunden>,<Auslöser>;. Der Auslöser setzt dabei Animationen in Gang, die natürlich vorher definiert sein müssen.

Hier werden also gleich drei Themen im Umgang mit Traffic angesprochen: 1.Bewegungen, 2. Animationen und 3.Modifikationen. Die Themen Animationen und Modifikationen werden in einem anderen Kapitel ausführlich behandelt.

Die folgenden Beispiele werden also eine ganze Reihe von Animationen und Modifikationen beinhalten, um die Möglichkeiten von Traffic zu veranschaulichen. Die zusätzlichen Parameter, die jede Bewegungszeile enthalten muss, wenn man eine individuelle Trasse nutzt werden als erstes beschrieben.

Diese Parameter betreffen die Höhe der Fahrzeuge und der Trasse. Normalerweise ist ein Fahrzeug nicht höher als 58 Pixel, und das ist auch die Vorgabe bei Traffic. Nun gibt es aber Fahrzeuge (z.B. Bahndienstfahrzeuge mit Kran), die deutlich höher sind. Dann muss man mit dem Parameter IMGY=; die Höhe angeben, sonst bleibt die Bewegung unsichtbar. Einige dieser Abmessungs-Parameter werden üblicherweise im Fahrplankopf als genereller Fahrplanbefehl angegeben, weil sie die Darstellung, den Platz und die Anzahl der Bewegungszeilen grundsätzlich regeln. Die Abmessungs-Parameter können aber auch jeder Bewegungszeile einzeln zugeordnet werden und somit die Vorgabewerte im Fahrplankopf überschreiben.

ABOVE = <Simple>;	Maximale Höhe des Vorder- und Hintergrundbildes oberhalb des Fahrzeugs.
BELOW = <Simple>;	Maximale Höhe des Vorder- und Hintergrundbildes unterhalb des Fahrzeugs.
IMGY = ; <Simple>	Höhe des Fahrzeugs. Muss nur angegeben werden, wenn es mehr als 58 Pixel sind.

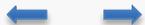
Die zusätzlichen Parameter, die jede Bewegungszeile enthalten kann, aber nicht zwingend sind, werden hier beschrieben:

A = <Acceleration>;	Beschleunigungsrate des Zugs. Zahl mit Punkt zur Trennung (Beispiel 0.4)
B = <Deceleration>;	Verzögerungsrate des Zugs. Zahl mit Punkt zur Trennung (Beispiel 0.4)
BG= <FgBg>;	Hintergrund. Können mehrere Bilder und Animationen sein, die in Ebenen und Positionen angeordnet werden.
D = <Direction>;	Fahrtrichtung, in der sich Fahrzeuge bewegen. Ohne Angabe sind es beide Richtungen.
FG= <FgBg>;	Vordergrund. Können mehrere Bilder und Animationen sein, die in Positionen und Ebenen angeordnet werden.
P = <Place>;	Haltepunkt der Fahrzeuge. Positionsangaben werden mit Satzzeichen (, < * - +) und Zahlenwerten bestimmt.
T = <WaitTime>;	Wartezeit, während der Zug steht, in der Animationen aufgerufen werden können. Beispiel: (T=1,w,HP1,w);
TB= <WaitTime>;	Wartezeit und Aktionen bevor der Zug auf dem Bildschirm erscheint.
TE= <WaitTime>;	Wartezeit und Aktionen nachdem der Zug den Bildschirm verlassen hat.
TBG=<Temporary Background>;	Ein Hintergrundbild, das durch das vorbeifahren der Fahrzeuge „aufgegessen“ wird, also danach nicht mehr zu sehen ist. Kann man gut für Schnee, Schotter und auch Fahrgäste auf einem Bahnsteig nutzen.
TX= <LineText>;	Beschreibungstext. Jede Bewegungszeile kann einen begleitenden Text haben.
TXC= <Color>;	Buchstabenfarbe des Beschreibungstextes.
TXX= <Place>;	Horizontale Position des Beschreibungstextes.
TXY= <Integer>;	Vertikale Position des Beschreibungstextes.
TXB= <Color>;	Hintergrundfarbe des Beschreibungstextes.
V =<Speed>	Geschwindigkeit mit der sich die Fahrzeuge bewegen.
WP=<WayPoint>	Wegpunkte auf der Trasse, die Animationen auslösen können. Beispiel: (WP=20,HP0;)
W = <Row>;	Vordefinierte Trassen, die statt der Hintergrundbilder benutzt werden können.
XBG=<Growable Background>;	Das Gegenteil von TBG=; Hier wird nach dem vorbei fahrenden Zug ein Hintergrund gezeichnet. Es funktioniert in jeder Fahrplanzeile aber nur entweder XBG=; oder TBG=; nicht beides zusammen.

HINWEIS:

Die Beispiele im Folgenden sind dazu gedacht, sie einzeln in einen Test-Fahrplan zu kopieren und auszuprobieren. Damit die Darstellung der Beispiele richtig funktioniert, sollten im Kopf des Testfahrplans Angaben zur Fahrwegtrasse gemacht werden. Werden generell eigene Trassen genutzt, ist zu empfehlen, grundsätzlich folgende Zeilen in den Fahrplankopf zu schreiben:

\$BGCOLOR #000000	Hintergrundfarbe des Bildschirms (falls die Fahrzeugliste eine andere Hintergrundfarbe hat).
\$IV 0.5	Die Anfangsgeschwindigkeit nach einem Halt (wird durch diese Angabe weicher).
\$ABOVE 30	Die Höhe der Trasse über den vorgegebenen 58 Pixeln.
\$BELOW 10	Die Höhe der Trasse unterhalb der Trasse.
\$WAY A;E;M;R;T;L=X	Schließt die Standard-Trassenparameter durch das X aus.
\$IMGY= 100	Fahrzeughöhe damit, Fahrzeuge die höher als 58 Pixel dargestellt werden (sonst bleibt die Zeile leer!)



Die Beschreibungen der Beispiele werden genutzt, um Trassendefinitionen und die Anordnung und Positionierung von Hinter – und Vordergrundelementen zu erläutern. Mit Klick auf die einzelnen Bewegungen gelangt man zu den Beispielen. Man kann aber auch einfach „scrollen“.

M=NULL	M=M0	Keine Bewegung. Wird genutzt um Logos und Beschreibungen anzuzeigen.
M=SIMPLE	M=M1	Einfache Bewegung- entspricht dem C=; (M=SIMPLE; Braucht nicht angegeben werden).
M=FOLLOW	M=M2	Mehrere Fahrzeuge folgen in einem mehr oder weniger dichten Abstand aufeinander.
M=STOP	M=M3	Fahrzeug hält. Nach einer Wartezeit, fährt es in der gleichen Richtung weiter.
M=BACK	M=M4	Fahrzeug hält. Nach einer Wartezeit, fährt es in der Gegenrichtung weiter.
M=CUT	M=M5	Fahrzeuge werden nach einer Wartezeit getrennt, und fahren getrennt weiter.
M=CHANGE	M=M6	Fahrzeuge werden mit Hilfe eines Dritten Fahrzeugs ersetzt (z.B. Lokwechsel).
M=OPEN	M=M7	In der Wartezeit werden Türen geöffnet und wieder geschlossen. (Benötigt eine DOOR Animation)
M=UNIT	M=M8	Fahrzeuge werden nach einer Wartezeit verbunden, und fahren vereint weiter.
M=PUT	M=M9	Ein oder mehrere Fahrzeuge bekommen durch ein Drittes, ein oder mehrere Fahrzeuge dazu.
M=GET	M=M10	Ein oder mehrere Fahrzeuge bekommen durch ein Drittes, ein oder mehrere Fahrzeuge abgenommen.
M=ECHG	M=M11	Im Prinzip wie M=6. Nur mit vier Fahrzeugbewegungen statt drei. (Kann universell genutzt werden)
M=HEAD	M=M12	Im Prinzip wie M=10. (Wird als Kopfbahnhof-Ankunft-Animation genutzt)
M=PUSH	M=M13	Im Prinzip wie M=6. (Wird als Kopfbahnhof-Abfahrt-Animation genutzt)
M=CUT2	M=M14	Im Prinzip wie M=5. Nur das in der Wartezeit aufgeteilt ist, um zusätzliche Animation vorzunehmen.
M=TURN	M=M15	Das erste Fahrzeug von mehreren wird an das Ende gesetzt. (Kopf machen)
M=RCUT	M=M16	Fahrzeuge werden während der Fahrt voneinander getrennt. (z.B. Abrollberg)
M=PASS		Ein oder mehrere Fahrzeuge halten. In der Wartezeit fahren andere Fahrzeuge davor vorbei.
M=MEET		Ein Zug hält. Nach einer Wartezeit hält ein zweiter Zug davor, wartet und fährt weiter. Der erste Zug fährt danach. (Bahnhofs-Situation)

M=NULL;M=M0;

Eine Bewegung, die eigentlich gar keine wirkliche Bewegung ist, da sie über keinen C= ; Parameter verfügt, der einen Zug definiert. Sie zeigt lediglich einen Hintergrund auf dem Bildschirm an. Diese Bewegung ist vorgesehen, um Logos oder Beschreibungen für die Folgezeilen anzuzeigen.

Die Zeitdauer der Anzeige wird über die Parameter TB =; und TE =; festgelegt. Es ist auch möglich, Animationen im Hintergrund zu starten.

Beispiel:

```
M=M0;BG=E:\Traffic\stock\BgPic\DARU_tr.gif;AB=500;BL=400;TB=w,2;TE=w,4;
```

Es ist erforderlich M=M0; zu schreiben. Bei M=NULL; bleibt der Bildschirm schwarz. Das anzuzeigende Bild muss nicht zwingend in der Fahrzeugliste vorhanden ist. Eine vollständige Pfadangabe geht ebenso.



M=SIMPLE;M=M1;

Einfache Vorbeifahrt. Dieser Wert ist der Vorgabewert für Bewegungen. Es ist die einfachste Form der Bewegung C=<Train>; In diesem Falle ist es nicht notwendig, die Bewegung mit M=SIMPLE; in der Fahrplanzeile explizit anzugeben. Es reicht ein C=;

Beispiel:

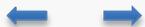
```
$DEF F_DRAHT=FL_DUF1,FL_DUF2|FL_DUF3
$DEF Trasse1=[BG=2*BUSH20:-20-40+,-7;BUSH20:60-80+;F_DRAHT:..., -3,#;TRBED25:...,^0][V=8-12]

$DEF F_DRAHT_SCH=3*(FL_DUF1,FL_DUF2,FL_DUF1,FL_DUF1)
$DEF Schranke=[BG=2*BUSH20:-20-40+=>SCH_BASIS,-7;BUSH20:SCH_BASIS+600,-7;F_DRAHT_SCH:SCH_BASIS-700,-3;\
  CROSS_H1:SCH_BASIS+40;PIKE4:SCH_BASIS+182)\
  [FG=TRBED25:...,^0;CROSS_H2:SCH_BASIS+330,0;CROSS_ROAD3:SCH_BASIS+210,-7;PIKE5:SCH_BASIS+231,0]\
  [TB=ON,2,w,BG.DOWN,w,FG.DOWN,w,OUT,w,2,w][TE=BG.UP,w,FG.UP,w,OFF,w,IN,2,w][V=7-10]
C=DB_3A_VT95_S1L,DB_3A_VT95_S1;AP=10:Trasse1|1:Schranke
```

In diesem Beispiel wurden zwei Trassen-Definition erstellt, die am Ende der Bewegungszeile mit AP=10xTrasse1 oder 1xSchranke aufgerufen werden. Die Definition der Trasse1 beinhaltet ein paar Büsche eine Oderleitung und ein Gleis, sowie die Angabe einer vergleichbaren Geschwindigkeit.

Die Positionierung der Schranke ist recht aufwendig und muss wegen der fünf Einzelteile und der Aufteilung in BG=; und FG=; genau sein, da sonst die Animation nicht richtig funktioniert. Die flexible Position von BUSH20 wird für die Positionierung der Schranke genutzt, damit nicht immer das gleiche Erscheinungsbild zu sehen ist, und keine Büsche auf dem Bahnübergang sind. Mit BUSH20:-20-40+=>SCH_BASIS wird ein Orientierungspunkt definiert, an dem dann alle Teile der Schranke, mit SCH_BASIS diesem Orientierungspunkt zugeordnet werden können. Auch die Oberleitung muss in diesem Fall an SCH_BASIS ausgerichtet sein, da sonst möglicher Weise ein Mast auf der Straße steht.

Die Auslöser der Schrankenanimation, werden in den Wartezeiten TB=; und TE=; mit BG.DOWN FG.DOWN usw. einzeln angesprochen. Da der Ausdruck sehr komplex ist, wird der gesamte Ablauf in die Definition \$DEF=Schranke geschrieben. Zum Aufruf der Trasse genügt dann ein AP=Schranke am Ende der Fahrplanzeile.





M=FOLLOW;M=M2;

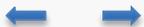
Mehrere Fahrzeuge folgen in kurzer Folge aufeinander. Die Fahrzeuge werden nicht aneinanderstoßen, selbst wenn sie unterschiedliche Geschwindigkeiten besitzen. Fahrzeuge müssen in dieser Bewegungsart durch Semikolon voneinander getrennt werden. Daher müssen sie insgesamt in Klammern eingeschlossen werden, da ein Semikolon normalerweise die einzelnen Parameter voneinander trennt. Also: C=(Tram, Wagen; Tram, Wagen1, Wagen2; Tram);

Beispiel:

```
$DEF Wohnhaus=D_H011|D_H011A|D_H011B|D_H011C|D_H011D|D_H012|D_H012A|D_H012B|D_H012C|D_H012D
M=FOLLOW;C=(BOCH_5GT6,BOCH_5GT6;BOCH_6GT6;BOCH_4GT6,BOCH_4GT6,BOCH_4GT6);\
BG=(20*Wohnhaus:22-70+;FL_TRAM:...,5,#;TRBED25:...,^0);AB=100;
```

Diese Fahrzeugbewegung funktioniert an sich am besten wenn keine zusätzlichen Parameter angegeben werden, weil schnellere Fahrzeuge, die folgen, abbremsen müssen. Gibt man mit V=; die Geschwindigkeit vor, gilt diese für alle Fahrzeuge, und der Bremsseffekt geht verloren. Die Fahrzeuge müssen in der Liste alle einzeln aufgezählt werden (2* BOCH_5GT6 geht leider nicht). Die Wohnhäuser im Hintergrund sind höher als die Vorgabe im Fahrplankopf. Deshalb wird mit AB=100; diese Vorgabe überschrieben.

Für Eisenbahnfahrzeuge ist diese Bewegung ungeeignet, da diese in Blockstrecken unterwegs sind. Für Trams und Autos hingegen – Genau das Richtige. Je mehr Fahrzeuge in einer C=; Definition sind, umso lustiger wird der Stau.



M=STOP;M=M3;

Ein oder mehrere Fahrzeuge halten. Nach einer Wartezeit fahren sie in der gleichen Richtung weiter.

Diese Bewegung hat (genau wie M=BACK; und M=SIMPLE) nur ein C=; Deshalb sind diese „einfachen“ Bewegungen durchaus geeignet, sie in eine Trassen-Definition aufzunehmen. In der Trassen-Definition lassen sich auch BG=; Ausdrücke flexibel gestalten, so das nicht immer das gleiche Erscheinungsbild beim Halten der Fahrzeuge zu sehen ist. Das gilt natürlich auch für Haltepunkte.

Teilt man die Fahrtrichtung noch in D=L; und D=R; auf, dann kann man, wie im folgenden Beispiel, eine fast allgemeingültige Trassen-Definition erstellen.

Beispiel:

```

$DEF Oberleitung=FL_DUF1,FL_DUF2|FL_DUF3
$DEF Busch=(BUSH38,BUSH39,BUSH40,BUSH41)|(#BUSH38,#BUSH39,#BUSH40,#BUSH41)
$DEF Gleis=TRBED24|TRBED25
$DEF Zaun=HANDRAIL07A,10-30*HANDRAIL07B,(0-1)*HANDRAIL07C,HANDRAIL07D
$DEF HALT_R=[BG=2-3*Busch:10-80+,-1;Zaun:5-95+;Oberleitung:...,#,\
[FG=Gleis:...,^0;SIG_DBHP:65-95+>=>Signal]][P=>Signal-40]\
[V=8-12][D=R][M=M3][T=2-3,w,FG.HP1,w,1]
$DEF HALT_L=[BG=2-3*Busch:10-80+,-1;Zaun:5-95+;Oberleitung:...,#,\SIG_DBHPL:5-40+>=>Signal]\
[FG=Gleis:...,^0][P=<Signal+40][V=8-12][D=L][M=M3][T=2-3,w,BG.HP1,w,1]
$DEF HALTEN=HALT_R|HALT_L

C=DBAG_101_029W;AP=HALTEN
    
```

Zuerst werden die Bestandteile von \$DEF HALT= definiert, um Abwechslung zu schaffen. In \$DEF HALT_L= und \$DEF HALT_R werden diese Bestandteile dann mit flexiblen Häufigkeiten und Positionen angeordnet. Damit der Zug nicht an einer zufälligen Position vor oder hinter den Signalen hält, sondern immer 40 Pixel davor, nutzt man die flexible Position (65-95+) des Signals SIG_DBHP als Orientierungspunkt=>Signal. Die Position des haltenden Zugs wird dann mit P=<Signal+40 diesem Orientierungspunkt zugeordnet. Bei Links- und Rechtsfahrt müssen dann nur die Vorzeichen getauscht werden und statt einem + ein - geschrieben werden, damit der Zug auch tatsächlich davor und nicht dahinter hält. Die Wartezeit T=; wird genutzt, um mit BG.HP1 und FG.HP1 die Signale einzeln anzusprechen, da der Auslöser HP1 bei beiden Signalen der Gleise ist. In der \$DEF=HALTEN werden HALT_R und HALT_L gleichberechtigt zusammengefasst.

Die Bewegungszeile selbst, braucht dadurch lediglich ein AP= HALTEN am Ende. Auch das M=STOP; muss nicht angegeben werden, weil es bereits in der Definition mit [M=M3] angegeben wurde.

M=BACK;M=M4;

Ein oder mehrere Fahrzeuge halten an. Nach einer Wartezeit fahren sie in der anderen Richtung weiter.

Auch diese Bewegung hat (wie M=STOP; und M=SIMPLE;) nur ein C=; und gehört damit zu den „einfachen“ Bewegungen, die man in eine Trassen-Definition aufnehmen kann, um z.B. um einen „Wendezug“ in Links- und Rechtsfahrt zu unterscheiden.

Beispiel:

```
$DEF Oberleitung=FL_DUF1,FL_DUF2|FL_DUF3
$DEF Busch=(BUSH38,BUSH39,BUSH40,BUSH41)|(#BUSH38,#BUSH39,#BUSH40,#BUSH41)
$DEF Gleis=TRBED24|TRBED25
$DEF Zaun=HANDRAIL07A,10-30*HANDRAIL07B,(0-1)*HANDRAIL07C,HANDRAIL07D
$DEF WENDEN_R=[BG=2-3*Busch:10-80+,-1;Zaun:5-95+;Oberleitung:..,-3,#;\
  PLF_I1_1:70-75+,0;10*PLF_5D:65-70+,-4;SIG_DBHPL:40-45+=>Signal]\
  [FG=Gleis:..,^0;SIG_DBHP:90-95+=>Signal][P=>Signal-40]\
  [V=8-12][D=R][M=M4][T=2-3,w,PD,w,1,PBU,w,2-3,w,BG.HP1,w,3-4]
$DEF WENDEN_L=[BG=2-3*Busch:10-80+,-1;Zaun:5-95+;Oberleitung:..,-3,#;SIG_DBHPL:5-12+=>SignalH]\
  [FG=Gleis:..,^0;PLF_I1_1:30-35+,0;10*PLF_5D:25-30+,-4;SIG_DBHP:55-60+=>Signal][P=<SignalH+40]\
  [V=8-12][D=L][M=M4][T=2-3,w,PD,w,1,PBU,w,2-3,w,FG.HP1,w,3-4]
$DEF WENDEN=WENDEN_R|WENDEN_L
C=<SNCF_X76513_514L,SNCF_X76513_514M,SNCF_X76513_514;AP=WENDEN
```

Wie im Beispiel M=STOP; sind die Haltepunkte an Signalen orientiert. Es gibt aber zwei Variablen für Links- und Rechtsfahrt. Im T=; Parameter wird einmal das linke Signal mit BG.HP1 angesprochen, und das andere Mal das rechte Signal mit FG.HP1. Die Möglichkeit des spezifischen Ansprechens ist sehr nützlich, wenn man mehrere gleichlautende Auslöser hat, wie in diesem Fall „HP1“. Außerdem sind im T=; Parameter auch Pantograph-Animations-Auslöser enthalten (PD,w,1,PBU). Diese Auslöser setzen zunächst alle Pantographen mit PD (Pantograph down) ab. Nach einer Wartezeit (w) wird der hintere Stromabnehmer mit PBU (Pantograph BACK UP) aufgebügelt. Diese Auslöser ergeben keinen Fehler, auch wenn ein Fahrzeug keine Pantographen hat. Die Fahrplanzeile ist auch hier erstaunlich kurz und braucht keine Angabe von M=BACK; , weil das bereits in der Definition enthalten ist. Das AP=WENDEN am Ende ruft die doch recht komplexe Definition auf.

M=CUT; M=M5;

Die Züge C1 und C2 kommen gemeinsam und halten. Nach der Wartezeit T1 fährt einer davon ab. Nach der zweiten Wartezeit T2 fährt der andere. Diese Bewegung ist besonders geeignet für das Auftrennen mehrerer gekuppelter Einheiten. Diese Bewegung hat keine besondere Beeinflussung des Kupplungstyps C. Wenn man einen animierten Kupplungsvorgang darzustellen möchte, der automatisch abläuft, sollte die Bewegung M=CUT2; eingesetzt werden.

C1=	Der linke Zugteil.
C2=	Der rechte Zugteil.
D =	Richtung, aus der die beiden Zugteile C1 + C2 erscheinen.
D1=	Richtung, in die der linke Zugteil ausfährt.
D2 =	Richtung, in die der rechte Zugteil ausfährt.
T1=	Wartezeit vor Abfahrt des ersten Teils. Wenn die Züge über animierte Seitenvariationen verfügen, laufen diese rückwärts ab, nachdem die Wartezeit beendet ist.
T2 =	Wartezeit und animierte Aktionen von der Abfahrt des ersten Teils bis zur Abfahrt des zweiten Teils.
V =	Geschwindigkeit, mit der die Zugteile C1 + C2 auf dem Bildschirm erscheinen.
V1 =	Höchstgeschwindigkeit, auf die der linke Zugteil beschleunigt.
V2 =	Höchstgeschwindigkeit, auf die der rechte Zugteil beschleunigt.

Beispiel:

```
$DEF Busch=(BUSH38,BUSH39,BUSH40,BUSH41)|(#BUSH38,#BUSH39,#BUSH40,#BUSH41)
$DEF Bahnsteig=PLF_NL7[O:LAMP19,100,10][O:LAMP19,800,10][O:BANK01,200,12][O:BANK01,600,12]
$DEF TR1T=[BG=2-3*Busch:5-40+,-1;2-3*Busch:70-100+,-1;CABLE2:...,-3,#;Bahnsteig:45-60+,-3;TRBED24:...^0][V=8-12]
M=CUT;C1=CFR_78_757[N:B,CONNECT1];C2=CFR_78_764;\
TB=1,DIESELRUN;P=@+(60-65+);T1=1,DIESELOFF,w,B>,w,1,w,2,C1.DIESELON,1,w,2,C2.DIESELON;\
T2=1,w,DIESELSTART,w,2;AP=TR1T
```



CONNECT1



Diesel3

Die Malaxa-Triebwagen haben eine in der Fahrzeugliste definierte eine „Diesel-Qualm-Animation“, die mit TB=DIESELRUN; ausgelöst wird. Außerdem wird in den Wartezeiten diese Diesel-Animation einzeln mit C1. und C2. und mehrfach mit OFF, ON und START variiert. Der Triebwagen CFR_78_757 bekommt eine zusätzliche Kuppel-Animation [N:B, CONNECT1]. Das Phasenbild CONNECT1 hat eine in der Fahrzeugliste definierte Animation, der man nun mit [N:], die ebenfalls definierte „Nachbar-Modifikation“ zuweist. Der Auslöser „B“ ist ein Kupplungstyp (es gibt auch C), der in T1=; mit B> aufgerufen wird.

HINWEIS:

Die Funktionen im Einzelnen werden im Kapitel „Animationen und Makros“ beschrieben.

M=CHANGE;M=M6;

Der Zug C1 und Zugteil C halten. Nach der Wartezeit T1 fährt C1 ab und lässt den Zugteil C stehen. Nach der Wartezeit T2 kommt Zug C2 und koppelt an den Zugteil C an. Nach der Wartezeit T3 fahren C2 und Zugteil C gemeinsam ab. Diese Bewegung ermöglicht einen Lokwechsel und/oder den Austausch von einem oder mehreren Wagen am Kopfende eines Zugs.

C =;	Der Zugteil, der während der gesamten Dauer der Bewegung auf dem Bildschirm bleibt.
C1 =;	Der Zugteil, der ausgetauscht wird.
C2 =;	Die Fahrzeuge, mit denen der Zugteil C1 ersetzt wird.
T1 =;	Wartezeit zwischen dem Anhalten von C1 + C bis C1 abgekuppelt losfährt. Wenn C1 und C über animierte Seitenvariationen im gekuppelten Zustand verfügen während T1 abläuft, wechseln diese in den ungekuppelten Zustand. Natürlich können diese Animationen auch innerhalb von T1 gestartet werden, wenn eine spezielle Reihenfolge benötigt wird oder Wartezeiten zwischen oder nach ihnen erforderlich sind.
T2 =;	Dauer und aufgerufene Animationen in der Zeit zwischen der Ausfahrt von C1 und dem Erscheinen von C2 auf dem Bildschirm.
T3 =;	Wartezeit zwischen dem Ankuppeln von C2 an C bis der neu formierte Zug wieder losfährt. Wenn es am Ende der Zeit irgendwelche animierten Seitenvariationen für den ungekuppelten Zustand gibt, gehen diese in den gekuppelten Zustand über.
T3C=;	Wartezeit vom Anhalten von C2 kurz vor dem Zug C bis zum Erreichen von C. Wenn C + C2 animierte Seitenvariationen des Typs C (Coupling) hat, die im ungekuppelten Zustand bei Ablauf der Zeit sind, werden diese in den gekuppelten Zustand wechseln.

Beispiel:

```
$DEF Capiale2=SNCF_Y_A9CAPL|SNCF_Y_A9CAP|SNCF_Y_A9RL|SNCF_Y_A9R
$DEF Capiale1=SNCF_Y_A9RNL|SNCF_Y_A9RN
$DEF WERBUNG=CFL_FOL_06|CFL_FOL_07_2|CFL_FOL_07_1|CFL_FOL_07|CFL_FOL_09|CFL_FOL_10_1|CFL_FOL_10|CFL_FOL_11|\
CFL_FOL_13|CFL_FOL_14_1|CFL_FOL_14|CFL_FOL_08|CFL_FOL_17|CFL_FOL_18_1|CFL_FOL_18|CFL_FOL_19_1|CFL_FOL_19
M=CHANGE;C=1-2*Capiale1,2-3*Capiale2;C1=SNCF_BB15000AL[>R];C2=SNCF_3_9200CML,SNCF_Y_A7DR;\
V=8-12;T1=1,w,2,w,2,w;T3=1,PD,w,2,PBU,2,w;P=@+(80-85+);BG=(FL_DUF1,FL_DUF2,FL_DUF3...,-3,#;PLF_11_1\
[O:WERBUNG,*0]:55-65+,0;15*PLF_5D:50-55+,-4;TRBED24:...,\^0);
```

In diesem Beispiel gibt es eine Overlay-Modifikation [O:WERBUNG,*0], die im Hintergrund auf dem Bahnsteig angebracht ist. Die Werbetafeln sind in der Definition \$DEF WERBUNG= mit einem | gleichberechtigt als „oder“ gelistet. Durch Komma getrennt wird zuerst die horizontale Position mit einem * (mittig, zentral) und danach mit der 0 die Höhe angegeben. Der Haltepunkt des Zugs wird mit einem @ für beide Fahrtrichtungen gleich, weil durch das @ die zurückgelegte Strecke gemessen wird und nicht der Abstand zum linken oder rechten Rand. Das [>R] hinter der Lok mit nur einem Pantographen bewirkt, dass die Lok immer rechts fahrend gezeigt wird, auch wenn sie nach links fährt. (Der Stromabnehmer ist immer hinten.)

M=OPEN;M=M7;

Ein Zug hält. In der Wartezeit werden die Türen geöffnet und wieder geschlossen. Der Zug fährt in der gleichen Richtung weiter. Diese Bewegung eignet sich **nicht** zum Aufteilen und Ansprechen einzelner Parameter in der Wartezeit. Möchte man in der Wartezeit weitere Animationen aufrufen eignet sich M=STOP besser. Im Kapitel **Modifikationen** gibt es dazu ein Beispiel mit Fahrgästen.

Möchte man allerdings nur Türen öffnen und schließen, funktioniert diese Bewegung am besten ohne Angabe von Parametern.

M=UNIT;M=M8;

Der erste Zug C1 hält. Danach kommt Zug C2 und kuppelt an den ersten an. Anschließend fahren beide Züge gekuppelt weiter.

C1 = ;	Fahrzeuge des ersten Zugs.
C2 = ;	Fahrzeuge des zweiten Zugs.
D = ;	Fahrtrichtung, in die der gekuppelte Zug ausfährt.
D1 = ;	Fahrtrichtung, aus der der erste Zug kommt.
D2 = ;	Fahrtrichtung, aus der der zweite Zug kommt.
T1 = ;	Wartezeit und Animationen nach dem Halt des ersten Zugs bis zum Eintreffen des zweiten Zugs auf dem Schirm.
T2 = ;	Wartezeit und Animationen nach dem Halt des zweiten Zugs bis zur Abfahrt der gekuppelten Einheit. Wenn animierte Seitenvariationen eines beliebigen Typs vorhanden sind, ändern sich diese am Ende des Zeitraumes in den gekuppelten Zustand.
T2C= ;	Wartezeit und Animationen zwischen dem Zeitpunkt, an dem C2 ein kleines Stück vor C1 anhält und bevor er an C1 ankuppelt. Wenn C1 und C2 über animierte Seitenvariationen vom Typ C verfügen, und die am Ende des Zeitraums im ungekuppelten Zustand dargestellt werden, ändern sich diese in den gekuppelten Zustand.
V= ;	Geschwindigkeit, die der gekuppelte Zug erreicht.
V1= ;	Geschwindigkeit, mit der der erste Zug auf den Bildschirm fährt.
V2 = ;	Geschwindigkeit, mit der der zweite Zug auf den Bildschirm fährt.

Beispiel:

```
$DEF ICE3=[PHU]<DB_403BF,DB_403BE,DB_403B1L,DB_403B1,DB_403B2L,DB_403WR,DB_403A1,DB_403AE,DB_403AF
M=UNIT;C2=ICE3;C1=ICE3;BG=(FL_DUF1,FL_DUF2,FL_DUF3:..,-3,#;PLF_5[BC:-0.2,0]:*,-3;TRBED24:..,^0);P=*;V=8-12;V1=8-12;V2=8-12;
```

Diese Bewegung ist das Gegenstück zu M=CUT2; . Es funktioniert sehr einfach und ohne Angabe zusätzlicher Parameter. Animierte Kuppel-Ereignisse werden automatisch gezeigt.

M=PUT;M=M9;

Wagen werden am Ende des Zugs angekuppelt. Der Zug C fährt ein und hält. C1 erscheint mit C2 von hinten und kuppelt C2 an C an. Dann verschwindet C1 wieder und C fährt anschließend zusammen mit C2 weiter.

C = ;	Fahrzeuge des Zugs.
C2 = ;	Die Rangierlokomotive.
C1 = ;	Die hinzuzufügenden Fahrzeuge.
T1 = ;	Wartezeit und Animationen zwischen dem Halt des Original-Zugs und dem Erscheinen der Rangierlokomotive.
T2 = ;	Wartezeit und Animationen zwischen dem Ankuppeln und dem Wegfahren der Rangierlokomotive. Wenn die Züge über animierte Seitenvariationen zwischen der Rangierlokomotive und den Extra-Wagen verfügen, gehen diese in die ungekuppelte Darstellung über. Diese Bewegung verfügt über keine spezielle Beeinflussung des Kupplungstyps C, weder zwischen der Rangierlokomotive und den Zusatzwagen noch zwischen dem Zug und den Zusatzwagen.
T3 = ;	Wartezeit und Animationen zwischen dem Wegfahren der Rangierlokomotive und dem Start des Zugs mit den angekuppelten Wagen. Wenn die Züge über animierte Seitenvariationen zwischen dem Zug und den Zusatzwagen verfügen, gehen diese in die gekuppelte Darstellung über.
T3C= ;	Wartezeit und Animationen zwischen dem Zeitpunkt, an dem C2 + C1 ein kleines Stück vor C anhält und bevor er an C ankuppelt. Wenn die Fahrzeuge über animierte Seitenvariationen vom Typ C verfügen, und die am Ende des Zeitraums im ungekuppelten Zustand dargestellt werden, ändern sich diese in den gekuppelten Zustand.
V = ;	Geschwindigkeit, mit der der Zug auf den Bildschirm fährt.
V1 = ;	Geschwindigkeit, mit der die Rangierlokomotive und die hinzuzufügenden Wagen erscheinen.
V2 = ;	Geschwindigkeit, mit der die Rangierlokomotive wegfährt.
V3 = ;	Geschwindigkeit des Zugs mit den angekuppelten Wagen.

Beispiel:

```
$DEF ANHANG=[N:B,CONNECT1]DB_4C_WGUGE841L|DB_4A_DUM902\
|DB_4B_DMS902BBL|DB_4A_DUE932KL|DP_4B_POST4C15|DB_POSTMRAP26_55L
M=PUT;C2=DB_4_260392;C=DB_3B_E101242,DB_4A_BUM231HL,DB_4A_BUM231H,DB_BDUMS282L,DB_4A_AUM201;
C1=ANHANG;BG=(FL_DUF1,FL_DUF2|FL_DUF3:..,-3,#;TRBED24:..,^0);P=@+(105-110+);T2=2,w,B<,w,2,B>,w,2,w;V=8-12;
```

Bei diesem Beispiel wird in der Definition `$DEF ANHANG=` die Kuppel-Animation `[N:B,CONNECT1]` vor die Liste der Wagen gestellt. Dadurch hat jeder Wagen, der in der Fahrplanzeile auftaucht, diese Animation. Durch das `@` Zeichen wird die Position des haltenden Zugs an der zurückgelegten Strecke gemessen. Das ist bei Bewegungen, die nicht in Links –und Rechtsfahrt aufgeteilt sind sehr praktisch. Diese Bewegung ist einfach zu realisieren, wenn man keine zusätzlichen Angaben macht, und ist das Gegenteil von `M=GET`;



M=GET;M=M10;

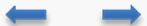
Wagen am Ende eines Zugs entfernen. Das Gegenteil der M=PUT; Bewegung.

C = ;	Der Zug, von dem Wagen entfernt werden - ohne die zu entfernenden Fahrzeuge.
C1 =;	Die abzukupelnden Fahrzeuge.
C2 = ;	Die Rangierlokomotive.
T1 = ;	Wartezeit und Animationen zwischen dem Halt des Original-Zugs C + C1 und dem Erscheinen der RangierlokC2.
T2 = ;	Wartezeit und Animationen zwischen dem Ankuppeln und dem Wegfahren der Rangierlokomotive C2 mit dem Zugteil C1. Wenn C1 und C animierte Seitendarstellungen haben, werden diese in den ungekuppelten Zustand versetzt. Diese Bewegung verfügt über keine spezielle Beeinflussung des Kupplungstyps C.
T3 = ;	Wartezeit und Animationen zwischen dem Wegfahren der Rangierlokomotive mit den abgekuppelten Wagen C1 + C2 und dem Start des Zugs C.
V = ;	Geschwindigkeit, mit der der Zug C + C1 auf den Bildschirm fährt.
V1 =;	Geschwindigkeit der Rangierlokomotive C2.
V2 = ;	Geschwindigkeit, mit der die Rangierlokomotive mit den und die abgekuppelten Wagen wegfährt C1 + C2.
V3 = ;	Geschwindigkeit des Rest-Zugs C.

Beispiel:

```
$DEF ANHANG=[N:B,CONNECT1]DB_4C_WGUGE841L|DB_4A_DUM902\
|DB_4B_DMS902BBL|DB_4A_DUE932KL|DP_4B_POST4C15|DB_POSTMRAP26_55L
M=GET;C2=DB_4_260392;C=DB_3B_E101242,DB_4A_BUM231HL,DB_4A_BUM231H,DB_BDUMS282L,DB_4A_AUM201;\
C1=ANHANG;BG=(FL_DUF1,FL_DUF2|FL_DUF3:..,-3,#;TRBED24:..,^0);P=@+(105-110+);T2=2,w,B<,w,2,B>,w,2,w;V=8-12;
```

Dieses Beispiel unterscheidet sich von dem Beispiel M=PUT; tatsächlich nur durch das Wort „GET“.



M=ECHG;M=SYSTEM_CHANGE; M=ECHANGE; M=M11;

Diese Bewegung ist vorgesehen, um eine Grenzstation mit zwei unterschiedlichen elektrischen Oberleitungssystemen nachzubilden, bei der es keine Möglichkeit der einfachen Umschaltung zwischen den Systemen gibt. Ein Zug hält. Eine Rangierlok erscheint und bringt die E-Lok fort, der restliche Zug bleibt ohne Lokomotive zurück. Danach erscheint eine dritte Lokomotive, kuppelt an den Zug an und der Zug setzt seinen Weg fort. Man kann diesen Vorgang natürlich auch zur Nachbildung anderer Vorgänge benutzen, beispielsweise zum Austausch einer defekten Lokomotive.

C = ;	Der Zug ohne die erste Lokomotive.
C1 = ;	Die E-Lok, die getauscht wird.
C2 = ;	Die Rangierlokomotive.
C3 = ;	Die neue Lokomotive.
T1 = ;	Wartezeit und Animationen zwischen dem Halt des Original-Zugs C + C1 und dem Erscheinen der Rangierlok C2.
T2 = ;	Wartezeit zwischen dem Ankuppeln und dem Wegfahren der C2 mit der alten Lok C1. Wenn die Züge über Seitenvariationen der ersten Lokomotive verfügen, laufen diese zum ungekuppelten Zustand ab. Animation an der Rangierlok laufen nicht automatisch ab.
T3 = ;	Wartezeit und Animationen zwischen dem Wegfahren der Rangierlok mit der ersten Lok C1 und der Ankunft der neuen Lok C3.
T4 = ;	Wartezeit und Animationen zwischen dem Ankuppeln der neuen Lok C3 und dem Wegfahren des Zugs. Alle Seitenanimationen zwischen der neuen Lokomotive und dem Rest des Zugs gehen am Ende der Wartezeit in den gekuppelten Zustand über.
V = ;	Geschwindigkeit, mit der der Zug C + C1 auf den Bildschirm fährt.
V1 = ;	Geschwindigkeit der Rangierlokomotive C2.
V2 = ;	Geschwindigkeit, mit der die Rangierlokomotive mit und die abgekuppelte Lokomotive wegfahren.
V3 = ;	Geschwindigkeit der neuen Lokomotive C3 bei deren Ankunft auf dem Bildschirm.
V4 = ;	Geschwindigkeit, mit der der neue Zug wegfährt C3 + C.

Beispiel:

```

$DEF Busch=(BUSH38,BUSH39,BUSH40,BUSH41)|(#BUSH38,#BUSH39,#BUSH40,#BUSH41)
M=ECHG;C=3-5*DRG_AB4UPR93CL,DRG_AB4UPR93C;C1=DRG_BAYES1[N:B,CONNECT1]; V=8-12;C2=DRG_V6001;\
C3=DRG_18_453[N:B,CONNECT1];BG=(3-4*Busch:0-50+;3-4*Busch:50-100+;A_BHF1D:60-70+,+4;10*PLF_5D[BC:-0.4,0]:60-65+,-3);\
FG=(FL_DUF3,FL_DUF2|FL_DUF1,#FL_DUFE:>70-75+=OB_Ende,-3;TRBED25:..,\^0);P=>OB_Ende-150;D=R;\
T1=w,1,PD,2,w,B<,2,w;T2=w,2,w,B>,2,w;T4=w,2,w,B<,w,2,w;
M=ECHG;C=3-5*DRG_AB4UPR93CL,DRG_AB4UPR93C;C1=DRG_BAYES1[N:B,CONNECT1]; V=8-12;C2=DRG_V6001;\
C3=DRG_18_453L[N:B,CONNECT1];BG=(3-4*Busch:0-50+;3-4*Busch:50-100+;A_BHF1D:25-30+,+4;15*PLF_5D[BC:-0.4,0]:50-55+,-3);\
FG=(FL_DUFE,FL_DUF3,FL_DUF2|FL_DUF1:>20-25+=OB_Ende_L..,-3;TRBED25:..,\^0);P=<OB_Ende_L+150;D=L;\
T1=w,1,PD,2,w,B>,2,w;T2=w,2,w,B<,2,w;T4=w,2,w,B>,w,2,w;
    
```

Dieses Beispiel wurde wegen der Positionierung des Oberleitungsendes in zwei Richtungen geteilt. Der Haltpunkt wird an **OB_Ende** und **OB_Ende_L** ausgerichtet.

M=HEAD;M=TERMINUS_ARRIVE; M=XBACK; M=M12;

Diese Bewegung stellt die Einfahrt eines Zugs in einen Kopfbahnhof dar. Eine andere Lokomotive (oder eine Rangierlokomotive) folgt nach und fährt mit den Wagen in die Gegenrichtung weg. Die Originallokomotive folgt nach. Diese Bewegung kann natürlich auch für andere Zwecke benutzt werden, so muss weder die "Rangierlokomotive" noch die "Originallokomotive" ein einzelnes Fahrzeug sein. Die Verwendung von Lokomotiven zusammen mit mehreren Wagen oder von Triebzügen ermöglicht weitere Effekte.

C = ;	Der Zug alleine - ohne die alte und neue Lokomotive.
C1 = ;	Die Original- (alte) Lokomotive.
C2 = ;	Die neue Lokomotive, die am Ende des Zugs ankuppelt.
T1 = ;	Wartezeit und Animationen zwischen dem Halt des Original-Zugs C + C1 und dem Erscheinen der neuen Lok C2.
T2 = ;	Wartezeit und Animationen zwischen dem Ankuppeln und dem Wegfahren der neuen Lok C2 mit dem Zug C.
T3 = ;	Wartezeit und Animationen zwischen dem Wegfahren der neuen Lok C2 mit dem Zug C + C2 und der Abfahrt der alten Lok C1.
V1 = ;	Geschwindigkeit, mit der der Zug C + C1 auf den Bildschirm fährt.
V2 = ;	Geschwindigkeit, mit der die neue Lokomotive ankommt.
V3 = ;	Geschwindigkeit der neuen Lokomotive mit dem Zug C + C2.
V4 = ;	Geschwindigkeit, mit der die alte Lokomotive wegfährt C1.

Beispiel:

```
M=HEAD;C=FS_6_UICX97_1L,FS_6_UICX97_1,FS_6_UICX97_2L,FS_6_UICX97_2;\
C1=FS_E655_177[N:B,KAPCSOL_A];C2=#CSD_T211_0[NR:#B,#T211POS];AB=80;\
BG=(PLF_I1_1,I_ORMEA[BC:.0.1,0];>75-80+100,0;BUSH39|BUSH38:>95-99+;#STREET3:>100+20,-6;\
FL_DUF3,FL_DUF2|FL_DUF1,#FL_DUFE->90-95+=>OB_Ende,-3;TRBED24:..<OB_Ende,^0;#BOCK2:OB_Ende-80,0);\
D=R;V1=3-5;V4=8-12;T1=w,1,B<,w,1,w;T2=w,1,PD,B>,1,w,2,PBU,w,2,w;P=>OB_Ende-100;
M=HEAD;C=FS_6_UICX97_1L,FS_6_UICX97_1,FS_6_UICX97_2L,FS_6_UICX97_2;\
C1=FS_E655_177[N:B,KAPCSOL_A];C2=#CSD_T211_0;\
BG=(L_STAZIONE_C,I_STAZIONE_C3,PLF_I1_1:30-35+100,0;BUSH39|BUSH38:0-10+;STREET3:0-20,-6;\
FL_DUFE,FL_DUF3,FL_DUF2|FL_DUF1:>5-7+=>OB_Ende_L,.,-3;TRBED24:..,^0;BOCK2:OB_Ende_L+80,0);\
D=L;V1=8-12;V4=8-12;T1=w,1,w,PD,w,B>,w;T2=w,1,w,B>,w,2,B<,w,PBU,w,2,w;P=<OB_Ende_L+100;
```

Das Oberleitungsende ist der Orientierungspunkt für die Position des Prellbocks und den Haltepunkt des Zugs. Der Bahnhof bei der Rechtsfahrt ist höher als die Vorgabe im Fahrplankopf, und deshalb benötigt diese Trasse die Angabe von **AB=80**;
 Die Bewegung ist das Gegenteil von M=PUSH;



M=PUSH; M=TERMINUS_DEPART; M=M13;

Eine Rangierlokomotive schiebt einen Zug in einen Kopfbahnhof, fährt rückwärts wieder weg und lässt den Zug dort stehen. Eine Streckenlokomotive kommt aus dieser Richtung, kuppelt an den Zug an und fährt mit diesem in Gegenrichtung weg.

Für Züge die in einen Kopfbahnhof einfahren benutzen Sie bitte die M=BACK; oder die M=HEAD; - Bewegung.

C = ;	Der Zug ohne Rangierlokomotive.
C1 = ;	Die Rangierlokomotive.
C2 = ;	Die Streckenlokomotive für den Zug.
T1 = ;	Wartezeit und Animationen zwischen dem Halt des Zugs C + C1 und dem Wegfahren der Rangierlokomotive C1.
T2 = ;	Wartezeit und Animationen zwischen dem Verschwinden der Rangierlokomotive C1 und der Ankunft der Streckenlokomotive C2.
T3 = ;	Wartezeit und Animationen zwischen dem Ankuppeln der Streckenlokomotive C2 an den Zug und der Abfahrt der neuen Kombination.
V1 = ;	Geschwindigkeit, mit der der Zug C + C1 auf den Bildschirm fährt.
V2 = ;	Geschwindigkeit, mit der die Rangierlokomotive wegfährt C1.
V3 = ;	Geschwindigkeit der Streckenlokomotive für den Zug C2.
V4 = ;	Geschwindigkeit, mit der die Streckenlokomotive mit dem Zug wegfährt C + C2.

Beispiel:

```
M=PUSH;C=FS_6_UICX97_1L,FS_6_UICX97_1,FS_6_UICX97_2L,FS_6_UICX97_2;\
C2=FS_E655_177[N:B,KAPCSOL_A];C1=CSD_T211_0;AB=80;\
BG=(I_ORMEA[BC:-0.1,0],PLF_I1_1:30-35+100,0;BUSH39|BUSH38:0-10+;STREET3:0-20,-6;\
FL_DUFE,FL_DUF3,FL_DUF2|FL_DUF1:>5-7+=>OB_Ende_L,-3;TRBED24:..,^0;BOCK2:OB_Ende_L+80,0);\
D=L;V1=3-5;V4=8-12;T1=w,1,B<,w,1,w;T3=w,1,PD,B<,1,w,2,PBU,w,2,w;P=<OB_Ende_L+120;\
M=PUSH;C=FS_6_UICX97_1L,FS_6_UICX97_1,FS_6_UICX97_2L,FS_6_UICX97_2;\
C2=FS_E655_177[N:B,KAPCSOL_A];C1=#CSD_T211_0[NR:#B,#T211POS];AB=80;\
BG=(PLF_I1_1,I_ORMEA[BC:-0.1,0]:>75-80+100,0;BUSH39|BUSH38:>95-99+;#STREET3:>100+20,-6;\
FL_DUF3,FL_DUF2|FL_DUF1,#FL_DUFE:>90-95+=OB_Ende,-3;TRBED24:..<OB_Ende,^0;#BOCK2:OB_Ende-80,0);\
D=R;V1=3-5;V4=8-12;T1=w,1,B>,w,1,w;T3=w,1,PD,B>,1,w,2,PBU,w,2,w;P=>OB_Ende-100;
```

Die Bewegung ist das Gegenteil von M=HEAD; Zu beachten ist, dass die Rangierlok [CSD_T211_0](#) in der Fahrzeugliste nur linksfahrend existiert. Bei der Rechtsfahrt wird die Lok mit einem # gespiegelt. Das wird auch mit der Kuppel-Animation so gemacht, die dadurch mit [NR:#B,#T211POS] extra genannt werden muss, da sonst die Animation nicht funktioniert.

M=CUT2; M=M14;

Die Züge C1 und C2 halten an. Nach der Wartezeit T1 entkuppelt einer davon und entfernt sich ein ganz kleines Stück. Nach der nächsten Wartezeit T1C fährt dieser ab. Nach der dritten Wartezeit T2 folgt der andere.

C1 = ;	Der linke Zugteil.
C2 = ;	Der rechte Zugteil.
D = ;	Richtung, aus der die beiden Zugteile C1 + C2 erscheinen.
D1 = ;	Richtung, in die der linke Zugteil ausfährt.
D2 = ;	Richtung, in die der rechte Zugteil ausfährt.
T1 = ;	Wartezeit und animierte Aktionen vom Halten bis zur kurzen Bewegung des ersten Teils. Wenn die Züge über animierte Seitenvariationen an ihrer Kupplungsseite (ausgenommen Typ "C") verfügen, laufen diese rückwärts ab, nachdem die Wartezeit beendet ist.
T1C=;	Wartezeit und animierte Aktionen zwischen der kurzen Bewegung und der Abfahrt des ersten Teils. Wenn die Züge über animierte Seitenvariationen an ihrer Kupplungsseite des Typs "C" verfügen, laufen diese rückwärts ab, nachdem die Wartezeit beendet ist.
T2 = ;	Wartezeit und animierte Aktionen von der Abfahrt des ersten Teils bis zur Abfahrt des zweiten Teils.
V = ;	Geschwindigkeit, mit der die Zugteile C1 + C2 auf dem Bildschirm erscheinen.
V1 = ;	Höchstgeschwindigkeit, auf die der linke Zugteil beschleunigt.
V2 = ;	Höchstgeschwindigkeit, auf die der rechte Zugteil beschleunigt.

Beispiel:

```
$DEF ICE3=[PHU]<DB_403BFL,DB_403BE,DB_403B1L,DB_403B1,DB_403B2L,DB_403WR,DB_403A1,DB_403AE,DB_403AF
M=CUT2;C2=ICE3;C1=ICE3;BG=(FL_DUF1,FL_DUF2,FL_DUF3:..,-3,#;PLF_5[BC:-0.2,0]:*,-3;TRBED24:..,^0);P=*;V=8-12;V1=8-12;V2=8-12;
```

Diese Bewegung ist das Gegenstück zu M=UNIT; Sie funktioniert sehr einfach und ohne Angabe zusätzlicher Parameter. Animierte Kuppel-Ereignisse werden automatisch gezeigt.

M=TURN; M=M15;

Ein Zug kommt an, die Lokomotive wird entkuppelt und wechselt auf einem Parallelgleis zur entgegengesetzten Seite, kommt von der anderen Seite zurück (dies kann alles im sichtbaren Bereich des Bildschirms stattfinden), kuppelt auf der Gegenseite des Zugs an und der Zug verlässt anschließend den Bildschirm in die Richtung, aus der er ankam.

C = ;	Der Zug ohne die Lokomotive.
C1 = ;	Die Lokomotive.
P = ;	Haltepunkt des Zugs. Ausrichtung Mitte bezeichnet den Kupplungspunkt zwischen Zug und Lokomotive.
P1 = ;	Erster Wendepunkt der Lokomotive. Die Bezeichnung Mitte bezieht sich auf die Mitte der Lokomotive.
P2 = ;	Zweiter Wendepunkt der Lokomotive. Die Bezeichnung Mitte bezieht sich auf die Mitte der Lokomotive.
T1 = ;	Wartezeit und Animationen zwischen dem Halt des Zugs und dem Start der Lokomotive.
T2 = ;	Wartezeit und Animationen, wenn die Lokomotive am ersten Wendepunkt anhält.
T3 = ;	Wartezeit und Animationen, wenn die Lokomotive am zweiten Wendepunkt anhält.
T4 = ;	Wartezeit und Animationen zwischen dem Ankuppeln der Lok an den Zug und der Ausfahrt des gesamten Zugs.
T4C = ;	Wartezeit und Animationen, wenn die Lok unmittelbar vor dem Ankupplungsvorgang kurz vor dem Zug anhält .
BEFORE;	Schlüsselwort ohne Parameter. Es spezifiziert, dass die Lok vor dem Zug zur Gegenseite wechselt.
BEHIND;	Schlüsselwort ohne Parameter. Es spezifiziert, dass die Lok hinter dem Zug zur Gegenseite wechselt. Natürlich kann nur ein Wert BEFORE oder BEHIND angegeben werden.
V = ;	Geschwindigkeit, mit der der Zug auf dem Bildschirm erscheint.
V2 = ;	Maximale Geschwindigkeit, die der weggehende Zug erreicht.

Beispiel:

```

$DEF Busch=(BUSH38,BUSH39,BUSH40,BUSH41)|(#BUSH38,#BUSH39,#BUSH40,#BUSH41)
$DEF Bahnsteig=(PLF2EL,PLF2P,PLF2N,PLF2T,5*PLF2P,PLF2E)[BC:-0.4,0]
M=TURN;C=2*DB_3A_BC3YG56L,DB_3A_BC3YG56;C1=DRG_2A_T11A[N:B,CONNECT1];BEFORE;D=R;V=6-8;\
FG=(WEICHE2:75-80+,-2;TRBED24:.,^0);P=<55-60+;T1=1,B< \
BG=(4-5*Busch:20-80+,-1;Bahnsteig:45-50+,-2;WEICHE2:15-20+,-2);T2=1,w,1,FG.LINKS,w,1,w;\
T3=1,w,1,BG.LINKS,w,1,w;T4=1,w,1,BG.RECHTS,w,1,w;
M=TURN;C=2*DB_3A_BC3YG56L,DB_3A_BC3YG56;C1=DRG_2A_T11A[N:B,CONNECT1];BEFORE;D=L;V=6-8;\
FG=(WEICHE2:75-80+,-3;TRBED24:.,^0);P=>60-65+;T1=1,B> \
BG=(4-5*Busch:20-80+,-1;Bahnsteig:45-50+,-2;WEICHE2:15-20+,-2);T2=1,w,1,BG.LINKS,w,1,w;\
T3=1,w,1,FG.LINKS,w,1,w;T4=1,w,1,FG.RECHTS,w,1,w;
    
```

Der Bahnsteig wurde mit der Modifikation BC insgesamt ein bisschen dunkler gemacht, und es gibt eine Weichenlaternen-Animation.

M=RCUT; M=M16;

Die Züge C1 und C2 fahren gemeinsam und werden während der Bewegung abgekuppelt.

Diese Bewegung verwendet keine animierten Seitenvariationen - die Seiten der Fahrzeuge bleiben beim Abkuppeln unverändert.

C1 = ;	Der erste Zugteil.
C2 = ;	Der zweite Zugteil.
P = ;	Entkopplungspunkt. Als Mittelpunkt der Einheit wird der Kupplungspunkt zwischen C1 und C2 angesehen.
V = ;	Geschwindigkeit, mit der die Kombination am Bildschirm erscheint.
V1 = ;	Geschwindigkeit, auf die der entkoppelte Zugteil beschleunigt.
V2 = ;	Geschwindigkeit, auf die der zweite Zugteil beschleunigt. (Kann bei einer Ablaufberg Situation auch langsamer sein)

Beispiel1:

```
M=RCUT;C2=DB_HABIS7TW3,DB_HABIS2,TWAG_HABIS2,BR_20132;\
C1=TWAG_HABIS3L;D=R;V=4;V1=5;V2=3;P=35;\
BG=(FL_DUF1,FL_DUF2,FL_DUF3:..,-3,#,TRBED24:..,^0);\
FG=(SIG_SLOW1:25,-2;WEICHE2:65-70+,-2);WP=(20,LINKS;81,RECHTS);
```

In diesem Beispiel wird die Situation des „Abstoßens“ dargestellt. Gute Ergebnisse erzielt man, wenn die Geschwindigkeiten zwischen den einzelnen Zugteilen nicht sehr groß sind.

Beispiel2:

```
$DEF FS_E404=FS_E404_661L,2*FS_ESAV_1,FS_ESAV_RIL|FS_ESAV_RI,4*FS_ESAV_2,FS_E404_661
M=RCUT;C2=NEUTRE20;C1=[PTU]<FS_E404;V=4;V1=18;V2=10;P=20;D=R;A1=0.1;\
BG=(FL_DUF1,FL_DUF2,FL_DUF3:..,-3,#,TRBED24:..,^0);FG=SIG_DBHP:50,-3;WP=15,HP1;TB=HP2,w,1;
M=RCUT;C1=NEUTRE20;C2=[PTU]<FS_E404;V=18;V2=8;P=5;D=R;B2=0.1;\
BG=(FL_DUF1,FL_DUF2,FL_DUF3:..,-3,#,TRBED24:..,^0);FG=SIG_DBHP:75,-3;TB=HP2,w,1;
```

Das zweite Beispiel ist eher ein „Trick“. Die Bewegung M=RCUT; ist außer M=FOLLOW; die einzige Bewegung die ein Abbremsen oder eine Beschleunigung während der Fahrt zulässt. Das kann man nutzen um z.B. einen schnellen Zug am P=; gezielt abzubremsen oder zu beschleunigen. Dazu ist es erforderlich, dass beim Bremsen C1 ein unsichtbarer Zug ist (NEUTRE20). Beim Beschleunigen ist C2 unsichtbar.

M=PASS;

Ein Zug C1 kommt und hält. Ein zweiter Zug C2 fährt an dem haltenden Zug im Vordergrund vorbei. Danach fährt C1.

Eine typische Vorbeifahrt Situation. Macht man hier keine Angaben zur Fahrtrichtung, dann bekommt man eine flexible zweigleisige Bewegung, da sowohl C1 und C2 in unterschiedlichen Richtungen fahren können.

C1 = ;	Der erste Zug, der auf dem Bildschirm erscheint und hält.
C2 = ;	Der zweite Zug, der vor dem ersten vorbeifährt.
P1 = ;	Haltepunkt des ersten Zugs.
V1 = ;	Geschwindigkeit des ersten Zugs C1, vor und nach dem Halt.
V2 = ;	Geschwindigkeit des vorbeifahrenden Zugs C2.
T1=;	Wartezeit nach dem Halt von C1.
T2=;	Wartezeit nachdem C2 den Bildschirm verlassen hat.

Beispiel:

```
$DEF Oberleitung=FL_DUF1,FL_DUF2|FL_DUF3
$DEF Busch=(BUSH38,BUSH39,BUSH40,BUSH41)|(#BUSH38,#BUSH39,#BUSH40,#BUSH41)
$DEF Gleis=TRBED24|TRBED25
$DEF Zaun=HANDRAILO7A,10-30*HANDRAILO7B,(0-1)*HANDRAILO7C,HANDRAILO7D
$DEF PASS=[BG=2-3*Busch:10-80+,-1;Zaun:5-95+;Oberleitung:...,-3,#;SIG_DBHP:65-75+=>Signal]\
[FG=Gleis:...^0;SIG_DBHP:85-90+][P=>Signal-40]\
[V1=4-5][V2=26-28][D1=R][T2=2-3,w,BG.HP1,w,1][TB=1,C1.1.FIRE,C1.3.FIRE,FG.HP1]
$DEF ICE3=<DB_403BFL,DB_403BE,DB_403B1L,DB_403B1,DB_403B2L,DB_403WR,DB_403A1,DB_403AE,DB_403AF

M=PASS;C1=EH_807,G_TORPEDOL,G_TORPEDOL,G_TORPEDOL;C2=ICE3;AP=PASS
```

In diesem Beispiel wurde wegen der animierten Signale die Fahrtrichtung von C1 festgelegt. Man beachte das Feuer bei den Torpedowagen. Diese Torpedowagen haben eine in der Fahrzeugliste bestimmte Animation „FIRE“. Im TB=; Parameter wird diese Animation mit **FIRE** ausgelöst. Allerdings nur bei zwei Wagen, denn mit Hilfe der Syntax **C1.1.** und **C1.3.** können bestimmte Fahrzeuge direkt angesprochen werden. Auch hier werden die Signale getrennt voneinander auf HP1 gesetzt.

M=MEET;

Ein Zug C1 erscheint und hält. Ein zweiter Zug C2 hält auf einem zweiten Gleis vor C1. Zug C2 fährt als erster wieder los. Zug C1 fährt danach.

C1 = ;	Der erste Zug, der auf dem Bildschirm erscheint.
C2 = ;	Der zweite Zug, der als erster wieder losfährt.
P1 = ;	Haltepunkt des ersten Zugs.
P2 = ;	Haltepunkt des zweiten Zugs.
V1 = ;	Höchstgeschwindigkeit, auf die der erste Zugteil beschleunigt.
V2 = ;	Höchstgeschwindigkeit, auf die der zweite Zugteil beschleunigt.
T1 = ;	Wartezeit nach dem Halten von C1.
T2 = ;	Wartezeit bevor C2 den Bildschirm verlässt.
T3 = ;	Wartezeit nachdem C2 den Bildschirm verlassen hat.

Beispiel:

```
M=MEET;C1=DB_V100_000MAK,DB_4A_BUE354KL,DB_4A_BUE354K,DB_41BYG514R;\
C2=DB_3B_E10,DB_4A_AUM201,DB_4A_BUM231GL,DB_4A_BUM231G,DB_4A_BUM231HL;\
V1=8;V2=8;D1=L;D2=R;\
FG=(PLF_5:40-55+,-4;TRBED24:.,^0;SIG_DBHP:<85-95+=>Signal);P2=>Signal-40;\
BG=(FL_DUF1,FL_DUF2,FL_DUF3:.,-3,#;PLF_5:40-55+,-4;SIG_DBHPL:<7-12+=>SignalH);\
T3=5,w,BG.HP1,w;T2=5,w,FG.HP1,2,w;P1=<SignalH+40;
```

Diese Bewegung eignet sich für eine Bahnhof-Situation, da die Darstellung zweigleisig ist. Auch hier sind die Haltepunkte an der Position der Signale orientiert. Da es zwei Signale sind, benötigt man auch zwei Orientierungspunkte (=>Signal und =>SignalH).

Um die Signalstellung HP1 unabhängig voneinander zu realisieren, werden die Signale im T=; Parameter mit **FG.HP1** und **BG.HP1** angesprochen.

Alle Bilder der Fahrzeugliste lassen sich durch Modifikationen verändern, erweitern und kombinieren. Das geschieht mit Hilfe von Makros. Es gibt eine ganze Reihe definierter Modifikations-Befehle, die grundsätzlich am Anfang innerhalb der Makro-Definition stehen. Dabei wird der gesamte Ausdruck in eckige Klammern vor oder hinter das betreffende Bild geschrieben. Das gilt für Fahrzeuge und Hintergrundbilder gleichermaßen. Es ist möglich diese Makros in die Fahrzeugliste als neues Makro aufzunehmen. Dadurch entsteht ein neuer oder veränderter Bildeintrag mit der entsprechenden Modifikation.

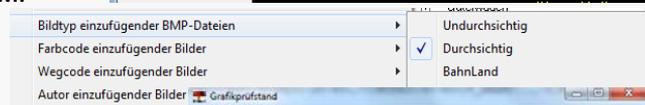
Es ist aber auch möglich, alle Makros in den Fahrplan zu schreiben ohne die Fahrzeugliste zu verändern. Bis auf wenige Ausnahmen bleibt die Funktion dabei die Gleiche. Es kommt also auf die Art der Veränderung oder Erweiterung an ob es sinnvoll ist, diese im Fahrplan, oder in der Fahrzeugliste vorzunehmen. Bei Fahrzeugen gibt es Kupplungen, Türen, Fenster, Räder, Pantographen usw. die fuhrzeugspezifisch sind, und deshalb einen Eintrag in der Fahrzeugliste rechtfertigen.

Bei Ladungen von Wagen hingegen, ist ein Fahrzeuglisteneintrag eher nicht sinnvoll, da diese Ladungen nicht immer die Gleichen sein sollen. Hier empfiehlt sich eine Makrodefinition im Fahrplan.

Die Möglichkeiten der Modifikationen sind nahezu unbegrenzt, deshalb wird hier nur an einigen Beispielen das Prinzip erklärt. Eine Liste aller Modifikations-Anweisungen befindet sich am Ende des Dokuments.

Die abgebildete E103 ist eine neue Zeichnung im 8bit BMP-Format. Sie wurde mit **Neu-> Bilddateien einfügen** in die Fahrzeugliste aufgenommen.

Dabei ist darauf zu achten, dass bei **Neu-> Bildtyp einzufügender BMP-Dateien** ein Häkchen bei **Durchsichtig** gesetzt ist, damit der schwarze Hintergrund der Lok auf dem Bildschirm transparent ist. Für den Einsatz im Fahrplan braucht die Lok aber noch eine Kupplungs- und Pantograph Animation, die in die Fahrzeugliste eingetragen wird.

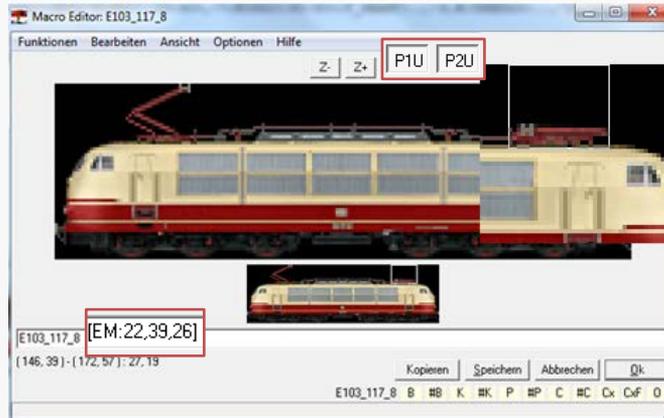


Es gibt in der Fahrzeugliste drei zusätzliche Fenster, von denen zwei im Erscheinungsbild nahezu identisch sind: Der Grafikprüfstand und der Makro Editor. Der Unterschied zwischen beiden ist die Möglichkeit den Eintrag zu speichern. Mit Rechtsklick auf die Lok kommt man zu **Makro -> Makro bearbeiten**. Es öffnet sich der Makro Editor, in dem die Modifikationen nun vorgenommen und gespeichert werden können. Ein gutes Hilfsmittel dazu, ist die Mausmenü-Datei die man unter Optionen eintragen kann, wenn nicht bereits geschehen. Sie heißt „**Traffic.gmm**“



und befindet sich im Stammordner von Traffic. Um eine einfache Pantograph Animation zu erstellen, müssen bei diesem Bild die Pantographen lediglich gespiegelt zu werden. Dazu zieht man mit gedrückter Maustaste ein Rechteck von links unten nach rechts oben, der den zu spiegelnden Bereich markiert.

Nun drückt man die rechte Maustaste und wählt den Eintrag [EM:] – Mirror pantographs. Es entsteht ein neuer Eintrag in der Makrozeile [EM:21,38,26], und es gibt zwei neue Schalter (Auslöser), mit denen man die Pantographen nun auf- oder abbügeln kann. Aber darum muss man sich nicht kümmern, denn das macht Traffic bei Rechts- und Linksfahrt selbst.

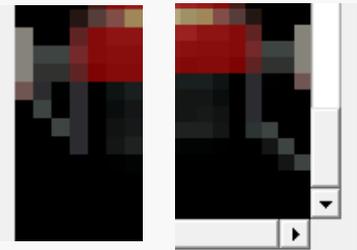
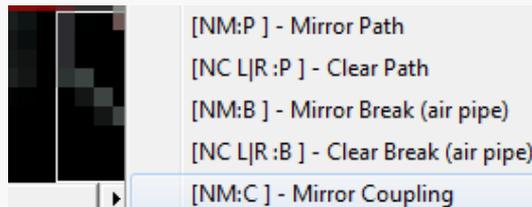


- Add to last used Window
- Koordinaten speichern
- [<L> MP:] - Mirror part
- [<L> MX:] - Mirror exchange
- [EM:] - Mirror pantographs

Diese E103 hat in der Zeichnung zwei unterschiedliche Kupplungszustände, was die Modifikation der Kupplung vereinfacht. Auch hier muss nur gespiegelt werden, um einen gekuppelten Zustand darzustellen.

Dazu wird wieder der zu spiegelnde Bereich markiert, und die Nachbar-Animation [NM:C] ausgewählt.

WICHTIG dabei ist, dass der jeweils **rechte** Teil der Zeichnung für die Spiegelung ausgewählt wird. Der in der Zeichnung dargestellte Zustand ist dabei nicht von Bedeutung, sondern die Größe des Auswahl-Bereichs, der bei dieser Art der Modifikation großzügig nicht so genau bemessen sein muss.



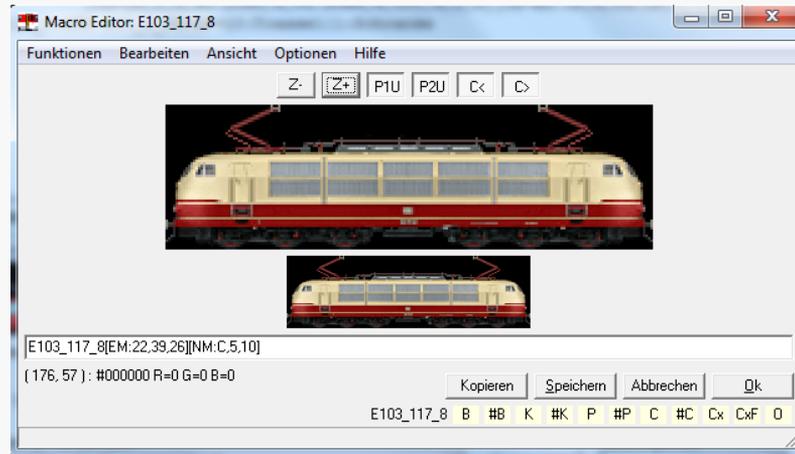
Das Ergebnis der mit Hilfe des Mausmenüs entstandenen Modifikation kann nun mit OK gespeichert werden.

Es lässt sich feststellen, dass diese Modifikationen ohne zusätzliche Bilder gemacht werden können, und der Ausdruck **nicht** selbst geschrieben werden muss.

So ist eine erweiterte Darstellung im Ablauf von Traffic möglich, ohne dass die Stellung der Pantographen oder der Zustand der Kupplung im Fahrplan erwähnt werden. Je nach Fahrtrichtung und vorhandenen Wagen, wird das jetzt von Traffic verwaltet.

Das Mausmenü lässt auch weitere Modifikationen zu. Je nach Art der Fahrzeuge lassen sich z.B. Türen mit einer DOOR Animation öffnen oder schließen. Auch das ist ohne zusätzliche Bilder möglich.

Zieht man mit der Maus **kein** Rechteck auf, sondern klickt an einer beliebigen Stelle mit rechts, dann bekommt man ein anders Kontextmenü zu sehen. Hier ist es dann möglich z.B. ein Blinklicht zu positionieren, was über dann über die Auslöser ON / OFF gesteuert werden kann.



Wie Anfangs bereits erwähnt, gibt es Modifikationen bei denen eine Definition im Fahrplan sinnvoll ist. Hier dient der Makro Editor oder der Grafikprüfstand dann zum Testen und Positionieren. Als Beispiel wird die Beladung eines Flachwagens (Ks) mit Holz beschrieben. In der Fahrzeugliste findet man diese unter **Güterwagen -> Flachwagen in Regelbauart, 2 achsig (K) -> Ks, Kbs**. Die Holzladung ist unter **Ladung -> Holz -> 2** zu finden.



Jetzt wird eine Auswahl der Holzstapel mit Hilfe der STRG-Taste getroffen und als \$DEF Holzstapel_Kurz =WOOD151|WOOD152|WOOD143|WOOD147|WOOD149|WOOD150 in eine Fahrplanzeile geschrieben. Die Modifikation ist in diesem Fall eine „Overlay behind“ [OB:x,y] Anweisung, da der Kbs 442 Rungen hat, die man sehen soll. Für die Positionierung der Holzstapel kann man den KBs Wagen im Makro Editor durch Doppelklick aufrufen.



Von Interesse sind hierbei nur die **x,y** Werte der Holzstapel, nicht die Holzstapel selbst. Man zieht dazu mit gedrückter Maustaste einen beliebigen Stapel auf die Position, an der der Stapel später zu sehen sein soll und lässt los. Diesen Vorgang wiederholt man, bis drei Stapel an der gewünschten Position über dem Wagen liegen. Die drei neu geschriebenen [O:] Anweisungen kopiert man mit STRG +C und fügt Sie als neue Zeile in den Fahrplan. Im Grafik Editor Fenster wählt man **Abrechen**, da sonst die Holzstapel in das Makro eingetragen Werden.



Im Fahrplan steht nun:

```
$DEF Holzstapel_Kurz=WOOD151|WOOD152|WOOD143|WOOD147|WOOD149|WOOD150 und in der neuen Zeile
DB_KBS442[O:WOOD151,18,16][O:WOOD151,54,16][O:WOOD151,90,16].
```

Zur Vervollständigung fügt man nun `$DEF Kbs_Holz_Kurz=` vor dem Wagen `DB_KBS442` ein. Jetzt wird `WOOD151` durch `Holzstapel_Kurz` ersetzt, und aus `[O:]` macht man `[OB:]`

```
$DEF Kbs_Holz_Kurz= DB_KBS442[OB:Holzstapel_Kurz,14,16][OB:Holzstapel_Kurz,52,16][OB:Holzstapel_Kurz,90,16]
```

Die vollständige Zeile beschreibt jetzt den Rungen-Wagen `DB_KBS442` mit unterschiedlichen Holzstapeln, die in der Bewegungszeile von Traffic ausgewählt werden, und somit ist immer ein anderes Erscheinungsbild des Wagens zu sehen.

Für einen gemischten Holz Zug braucht man noch lange Holzstapel, die die gesamte Ladefläche ausfüllen.

```
$DEF Holzstapel_Lang= WOOD186|WOOD187|WOOD16|WOOD54
```

Hier ist die Positionierung einfacher. Ein * positioniert mittig, und die Höhe (16) wird aus der ersten Definition übernommen.

```
$DEF Kbs_Holz_Lang= DB_KBS442[OB:Holzstapel_Lang,*,16]
```

Damit nicht nur ein Wagen das Holz transportiert wird auch hier eine Definition erstellt.

```
$DEF Kbs_Wagen=DB_KBS442|CFR_KS5|CFR_KS6
```

Diese Definition `Kbs_Wagen`, muss im Fahrplan natürlich vor den anderen Definitionen stehen, und dann `DB_KBS442` ersetzen.

In einem Beispiel-Fahrplan mit Lok sieht das dann so aus:

\$DEF **Kbs_Wagen** wurde um zwei Wagen erweitert, die als **Kbs_Wagen_Holz_K** und **Kbs_Wagen_Holz_L** mit Holz beladen werden können.

Hinter dem C= steht =[MI:9]. Diese globale Modifikation begrenzt die Anzahl der Fahrzeuge auf maximal neun.

Das wurde gemacht damit der Zug durch die verschachtelte Mengenangabe nicht zu lang wird. Fünf bis acht Wagen und das in Gruppen (@) von zwei bis vier Wagen, das kann insgesamt recht lang werden. Die Gruppierung betrifft nur die Wagen, nicht die Ladung. Dadurch werden interessante Züge gebildet.

Man muss also keine beladenen Züge zeichnen, sondern stellt sie mit [O:] zusammen. Auch bei Containerzügen ist das von großem Nutzen.

Eine besondere Form der Modifikation ist die Animation aus Phasenbildern. Da gibt es die Möglichkeit ein Fahrzeug aus mehreren Bildern zusammen zu setzen (Bild1+Bild2+Bild3). Diese Art der Animation funktioniert nur bei Fahrzeugen und ist dann abhängig von der gefahrenen Geschwindigkeit.

Bei der anderen Art der Phasenbild-Animation werden alle Phasen der Animation in einem Bild als Filmstreifen zusammengefasst. Der Ablauf der Animation erfolgt dann, indem jeweils eine Phase für eine bestimmte Zeit in einem „Frame“ dargestellt wird. Das ist sowohl bei Fahrzeugen, als auch im Hinter- und Vordergrund möglich. Dabei wird die Animation mit „Schaltern“ ausgelöst, die dann entweder in den Wartezeiten abläuft, oder an Wegpunkten (WP=). Die Möglichkeiten der Animation sind dadurch nahezu unbegrenzt. Es müssen nur die entsprechenden Filme „gedreht“ werden.

```
$BGCOLOR #000000
$IV 0.5
$ABOVE 30
$BELOW 10
$WAY A;E;M;R;T;L=X
$IMGY 100

$DEF F_DRAHT= FL_DUF1,FL_DUF2|FL_DUF3
$DEF Busch= BUSH20|#BUSH20
$DEF Trasse1= [BG=(Busch:<10-40+,-8;Busch:<60-80+,-8;F_DRAHT:...,-3,#;TRBED25:...,*0;)]\
[V=8-12]
$DEF Kbs_Wagen=DB_KBS442|CFR_KS5|CFR_KS6
$DEF Holzstapel_Kurz=WOOD151|WOOD152|WOOD143|WOOD147|WOOD149|WOOD150
$DEF Holzstapel_Lang= WOOD186|WOOD187|WOOD16|WOOD54

$DEF Kbs_Wagen_Holz_K=Kbs_Wagen[OB:Holzstapel_Kurz,14,16]\
[OB:Holzstapel_Kurz,52,16][OB:Holzstapel_Kurz,90,16]
$DEF Kbs_Wagen_Holz_L=Kbs_Wagen[OB:Holzstapel_Lang ,* ,16]

C=[MI:9]DB_V100_000MAK,5-8*(2-4@(Kbs_Wagen_Holz_K | Kbs_Wagen_Holz_L));AP=Trasse1
```

Die Flashdemo zeigt das Prinzip einer in Traffic definierten [A:PAN] Animation. In diesem Fall reicht es aus, den Namen des animierten Bildes, die Position und die Anzahl der Frames anzugeben, um eine Animation auszulösen.

[A:PAN,Bildname,x,y,FrameBreite]

Die Position des Frames auf der Lok wird als x, y in Pixeln angegeben.

Die Frame-Breite muß in Pixeln angegeben werden.

-Demo

auf ab

Zum Testen die Pfeile drücken.

Bei anderen Animationen, die nicht in Traffic vordefiniert sind, benötigt man mehr Angaben.

Als Beispiel kann man die folgende Fahrgastanimation in die Traffic-Fahrzeugliste einfügen. Sie ist im Ordner der Dokumentation zu finden. Das Phasenbild

[PASS1_ANI](#) besteht aus

20 Phasen, wobei jeder

„Frame“ eine Breite

von 33 Pixeln und eine

Höhe 22 Pixeln hat.

Dargestellt sind drei

Personen, die in einen

Zug „einsteigen“.

PASS1_ANI[ANIM:(Stehen=0;Einsteigen=1-19),33,22,0,0,15,20,33,0]

(0, 0) - (32, 21) : 33, 22

Kopieren Speichern Abbrechen OK

Durch Rechtsklick auf [PASS1_ANI](#) kommt man zu **Makro -> Makro bearbeiten**. Da wird dann das obenstehende Makro eingetragen und mit OK bestätigt. Es setzt sich wie folgt zusammen:

Der Modifikationsbefehl [ANIM: danach in Klammern die „Schalter STEHEN“ und „EINSTEIGEN“ mit = Frame Angabe.

Durch Komma getrennt, **Breite(33)**, **Höhe(22)**, x Position(0), y Position(0), **Dauer des gezeigten Frames in 10 m/sec(15)**, **Anzahl der Frames(20)**, **x Abstand zum nächsten Frame(33)**, y Abstand zum nächsten Frame(0)]

Damit die Fahrgäste auf einem Bahnsteig positioniert werden können, braucht man nun ein neues Makro, das man mit Rechtsklick auf das bereits definierte Makro [PASS1_ANI](#) erstellt. Der Name ist frei wählbar ([PASS1_M](#)) ist nur ein Vorschlag. Mit Klick auf OK öffnet sich ein neues Fenster „Makro Editor: PASS1_M“

Neues Makro

Name des neuen Makros

PASS1_M

Cancel OK

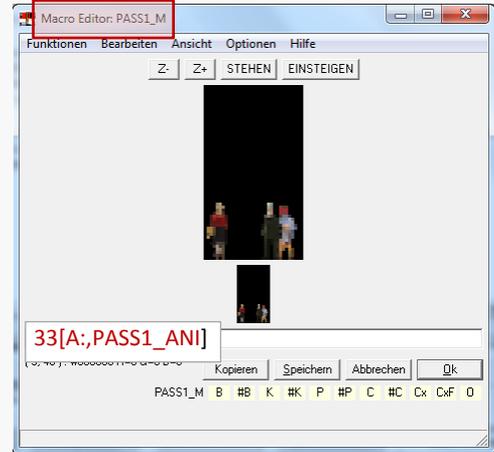
Dort sieht man zunächst die bereits existierende Makrodefinition. Das ändert sich wenn man `33[A:;PASS1_ANI]` schreibt. Die Zahl `33` beschreibt die Grundfläche der neuen Animation. Die Modifikationsanweisung `[A:;]` gefolgt von einem Komma, und dem Namen der bereits definierten Fahrgast-Animation „`PASS1_ANI`“ zeigt nun **einen** Frame mit den Auslösern „`STEHEN`“ und „`EINSTEIGEN`“.

Mit OK bestätigen und man erhält die Fahrgast-Animation `PASS1_M`.

Diese Animation muss nun an den Positionen der Türen eines haltenden Zugs angepasst werden, damit sie auch einsteigen können. Dazu gibt es mehrere Möglichkeiten.

Im Folgenden Beispiel wird die Position eines Signals der Orientierungspunkt für die Position das haltenden Zugs.

Die Fahrgäste bekommen dann die Position des Signals + Haltepunkt + Türmitte eines Wagens zugewiesen.



```

$DEF Fahrdraht= FL_DUF1,FL_DUF2|FL_DUF3
M=STOP;C=<FS_6_ETR425JAL,FS_6_ETR425JBL,FS_6_ETR425JCL,FS_6_ETR425JDL,FS_6_ETR425JE;V=8-12;D=L;\
BG=(Fahrdraht:..., -3, #, SIG_DBHPL:10-15+ =>SignalH, -5; TRBED24:..., ^0);\
P=<SignalH+40; FG=(10*PLF_5D:SignalH+450, -4; PASS1_M:SignalH+311, +4);\
T=1, C.2.OPEN, w, 1, EINSTEIGEN, w, CLOSE, w, 2, BG.HP1, w, 2;
    
```

Dadurch kann die Position des Signals: `10-15+` eine flexible Positionsangabe sein, die Fahrgäste stehen immer richtig. In der Wartezeit `T=`; wird nur die zweite Tür mit `C.2.OPEN` angesprochen. Die anderen Türen bleiben geschlossen.

Liste der Modifikationsbefehle

[#A:]	Beschleunigung	[EU:]	Stromabnehmer aufstellen
[#B:]	Verzögerung	[EX:]	Stromabnehmer ändern
[#V:]	Geschwindigkeit	[ET:]	Stromabnehmer bewegen transparent
[#W:]	Trassendarstellung	[F:]	Farbe füllen
[A:]	Animation)	[FL:]	Blinklicht
[AT:]	Transparente Animation)	[L:]	Bezeichner
[ANIM:]	Animation festlegen	[M:]	Spiegeln
[AO:]	Einfache Animation	[MP:]	Teil spiegeln
[ABC:]	Buchstabendefinition	[MX:]	Spiegeln und tauschen
[B:]	Block füllen	[MX1:]	Spiegeln und tauschen
[BC:]	Helligkeit und Kontrast	[N:]	Nachbarn
[BT:]	Block transparent füllen	[NAME:]	Name / (Name)
[C:]	Farbe	[NC:]	Nachbarverbindung löschen
[CB:]	Kuppeln	[NCL:]	Linke Nachbarverbindung löschen
[CL:]	Kuppeln links	[NCR:]	Rechte Nachbarverbindung löschen
[CR:]	Kuppeln rechts	[NL:]	Linker Nachbar
[CP:]	Kopieren	[NM:]	Nachbarverbindung spiegeln
[D:]	Definition	[NR:]	Rechter Nachbar
[DA:]	Türen bewegen	[NT:]	Nachbartyp
[DB:]	Definition hinter	[NTL:]	Nachbartyp links
[DID:]	Doppeltür innen	[NTR:]	Nachbartyp rechts
[DIL:]	Tür innen links	[O:]	Überdeckung
[DIR:]	Tür innen rechts	[OB:]	Überdeckung hinter
[DOD:]	Doppeltür außen	[OT:]	Transparente Überdeckung

[DOL:]	Tür außen links	[P:]	Punkt
[DOR:]	Tür außen rechts	[PB:]	Punkt hinter
[DT:]	Definition transparent	[PT:]	Punkt transparent
[E:]	Stromabnehmer bewegen	[R:]	Rückwärts
[EC:]	Stromabnehmer kopieren	[T:]	Text
[ECM:]	Stromabnehmer kopieren und spiegeln	[TL:]	Linker Nachbar transparent
[ED:]	Stromabnehmer absenken	[TN:]	Nachbarn transparent
[EM:]	Stromabnehmer spiegeln	[TR:]	Rechter Nachbar transparent
[EMC:]	Stromabnehmer spiegeln und kopieren	[TT:]	Nachbartyp transparent
[EMX:]	Stromabnehmer spiegeln und tauschen		

