

# Skripte für KS-Signale im Train Simulator (Version 9.0)

Im Zuge der deutschen Wiedervereinigung und der damit verbundenen Zusammenlegung der Deutschen Bundesbahn mit der Deutschen Reichsbahn entstand die Notwendigkeit, die in Ost und West unterschiedlichen Signalsysteme zu vereinheitlichen.

Das bei der Bundesbahn übliche H/V-Signalsystem und das bei der Reichsbahn verwendete HI-Signalsystem waren allerdings nicht zueinander kompatibel, sodass eine kurzfristige Umstellung aller Signale in einem Teil Deutschlands unmöglich war.

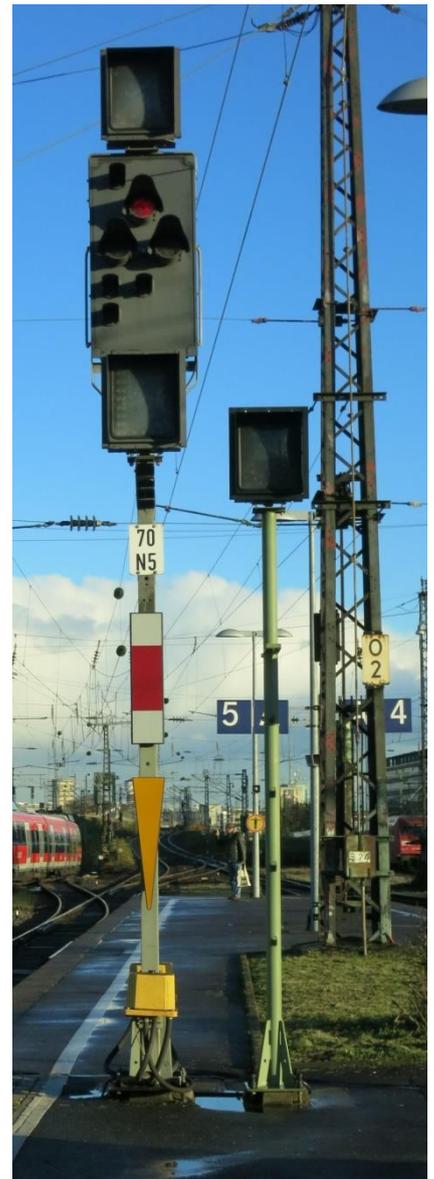
Die Vorstände der Bundes- und Reichsbahn beschlossen im März 1991 die Einführung dieses neuen Signalsystems. Es wurde dabei Ks-Signalsystem genannt, weil Vor- und Hauptsignal in einem Signal kombiniert werden konnten. Die Bezeichnung ist nicht mit den ebenfalls bei der Bundesbahn eingeführten Kompaktsignalen (KS) zu verwechseln, welche einen neuartigen Signalschirm für H/V-Signale darstellen und den Ks-Signalen optisch ähneln.

Quelle: Wikipedia

Sie können ihre selbst gebauten KS-Signale verschiedenster Bauformen unter Einsatz dieser Skripte auf Freewarestrecken im Train Simulator verwenden. Beachten Sie die Lizenzbedingungen unter Punkt 1.2.

Dieses Paket beinhaltet die Anleitung und Skripte, um selbst erstellte KS-Signale mit den Skript-Modulen aus dem Paket „**Freeware Skript-Module und Signal-Trigger**“ zu einem funktionierenden Signalsystem zu verbinden.

**Wenn Sie lediglich fertige Signale nutzen wollen**, die für die Schuster/Freeware - Skript-Module erstellt wurden, benötigen Sie dieses Paket nicht, sondern laden Sie sich von Rail-Sim.de das Paket „**Freeware Skriptmodule und Signal-Trigger**“ herunter und installieren dieses mit der Utilities.exe.



Bestandteile der Datei: Skriptpaket\_KS-Signale\_V9.0.zip

- Anleitung Skriptpaket\_KS-Signale\_V9.0.pdf
- Excel-Tabelle Code-Berechnung\_KS-Signale.xlsx
- Skripte für Source Skripte\_KS-Signale\_V9.0.zip

Indem Sie dieses Paket nutzen, akzeptieren Sie auch die Lizenzbestimmungen.

Bei Problemen/Anregungen bitte über das Forum [www.Rail-Sim.de](http://www.Rail-Sim.de) anschreiben.

Viel Erfolg beim Einsatz der Skripte für die KS-Signale wünscht  
Mathias Gundlach (Schuster at Rail-Sim)

## Inhaltsverzeichnis

<b>1. Einleitung</b> .....	<b>3</b>
1.1. Einige Vorworte .....	3
1.2. Lizenzbestimmungen .....	3
1.3. Installation .....	3
1.4. Neuerungen gegenüber der vorherigen Version .....	3
<b>2. Ordnerstruktur</b> .....	<b>4</b>
<b>3. Vorgaben für 3D-Modelle</b> .....	<b>4</b>
3.1. Allgemeines .....	4
3.1.1. Regeln für die Benennung der Licht-Nodes .....	4
3.1.2. Anmerkungen zu den Child-Namen .....	5
3.1.3. HP_CODE und VR_CODE .....	5
3.1.4. Zusatzanzeiger .....	6
3.2. Vorseignale .....	6
3.2.1. Vorseignal ohne Zusatzlicht .....	6
3.2.2. Vorseignal ohne Zusatzlicht mit Geschwindigkeitsvoranzeiger .....	6
3.2.3. Vorseignal im verkürzten Bremswegabstand .....	7
3.2.4. Vorseignalwiederholer .....	7
3.3. Hauptsignale (einige Beispiele) .....	8
3.3.1. Hauptsignal .....	8
3.3.2. Hauptsignal komplett mit Zs3 .....	9
3.3.3. Mehrabschnittsignal .....	9
3.3.4. Mehrabschnittsignal komplett mit Zs3 und Zs3v .....	10
3.3.5. Ausblenden von Signaloptiken .....	11
3.4. Sperrsignale .....	11
3.4.1. Sperrsignal ohne Deckungssignalfunktion .....	11
3.4.2. Sperrsignal und Deckungssignal .....	11
3.4.3. Wartesignal .....	12
3.5. Zusatzsignale und Zusatzanzeiger .....	12
3.5.1. Fahrtanzeiger .....	12
3.5.2. Abfahrtsignal .....	12
3.5.3. Richtungsanzeiger .....	13
3.5.4. Gleiswechselanzeiger .....	13
3.5.5. Geschwindigkeitsanzeiger .....	13
3.6. Signale ohne Signalschirm .....	14
<b>4. Skripte</b> .....	<b>14</b>
<b>5. BluePrintEditor</b> .....	<b>17</b>
5.1. Grundsätzliches .....	17
5.2. Köpfe .....	17
5.3. Einstellungen für Signale und Zusatzanzeiger .....	17
5.4. Export des BluePrints .....	20
5.5. Einsatz der Signale in einer Train Simulator Strecke .....	20

## 1. Einleitung

### 1.1. Einige Vorworte

Wer sich mit dem Bau von KS-Signalen beschäftigt, und diese Skripte verwenden möchte, sollte sich grundsätzlich mit dem Signalpaket Deutsche\_KS-Signale vom SignalTeam auseinandersetzen, um die Sonderfunktionen des Signalsystems zu verstehen.

Bezüglich der Sonderfunktionen im Zusammenhang mit Triggern und Einstellungen im Signal-Flyout unterscheiden sich diese Skripte nicht von denen, die unter „**Schuster/SignalTeam**“ veröffentlicht wurden. Hierbei ist jedoch der jeweilige Versionsstand zu beachten.

Trigger sind nicht Bestandteil dieses Skriptpaketes. Diese werden separat unter „**Schuster/Freeware**“ veröffentlicht und können dann für alle Signalsysteme dieser Veröffentlichungsreihe gemeinsam genutzt werden. Gleichfalls können auch alle bisher veröffentlichten Trigger aus anderen Signalpaketen, soweit sie dem aktuellen Stand entsprechen, genutzt werden.

### 1.2. Lizenzbestimmungen

Das Paket wird als Freeware auf Rail-Sim ([www.Rail-Sim.de](http://www.Rail-Sim.de)) angeboten und darf nicht auf weiteren Plattformen ohne meine Erlaubnis angeboten werden.

Die Skripte und Module dürfen ausschließlich auf **Freeware-Strecken** verwendet werden. Es ist nicht gestattet, die Skripte oder Module für kommerzielle Strecken zu verwenden.

Die Module dürfen nicht geändert, angepasst oder in anderen Provider-/Produktordnern gespeichert und dort heraus geladen werden. Die Module dürfen nicht mit Ihren erstellten Signalen gemeinsam veröffentlicht werden. Updates stelle ich ausschließlich selbst zur Verfügung.

Die Module dürfen nicht als Bestandteil von Strecken oder Signalpaketen verteilt werden. Sie dürfen nur per Link aus ihrer ursprünglichen Downloadquelle unter Rail-Sim angeboten werden.

Die im Klartext gelieferten Skripte dürfen angepasst, kopiert, exportiert und im eigenen Source-Ordner verwendet werden. Sie dürfen dann auch im OUT-Format innerhalb der eigenen Streckenpakete verpackt und verteilt werden.

Sollte Bedarf bestehen, dass diese Module in **Payware-Projekten** verwendet werden, so bitte ich um Kontaktaufnahme per E-Mail an [Railworks@mgundlach.de](mailto:Railworks@mgundlach.de). Die Module werden dann durch mich an den entsprechenden Provider- und Produktordner angepasst und eine Lizenz zur Verwendung vergeben.

### 1.3. Installation

Das Skriptpaket ist eine ZIP-Datei und muss mit einem geeigneten Programm entpackt werden. Die Skripte werden dann in den Source-Ordner kopiert, in dem später auch die eigenen Signale abgelegt werden.

### 1.4. Neuerungen gegenüber der vorherigen Version

#### Version 9.0

- Erste veröffentlichte Version des Skriptpaketes
- Gegenüber den vor 2019 veröffentlichten Freeware-Paketen ist es bei diesem Signalpaket auch möglich bei Haupt- und Mehrabschnittsignalen einzelne Signaloptiken per Code auszublenden.

## 2. Ordnerstruktur

- Hauptordner
  - **Assets\Schuster\Freeware\RailNetwork\Signals\German KS**

In diesem Ordner befindet sich lediglich die Optionsdatei. Diese muss dort unbedingt gespeichert bleiben und dient der Einstellung von einigen Optionen für das Signalsystem.

- Module

Der Ordner Module ist ein Unterordner von „German KS“ und enthält alle Skript-Module im OUT-Dateiformat, die aus den Skripten heraus nachgeladen werden. Diese Skriptmodule werden in keinem Blueprint direkt angegeben!

- Skripte

Im Bereich Source liegen in einem Unterordner der sinnvoller Weise „Skripte“ lauten sollte, unterhalb des eigenen Signalverzeichnisses alle Skripte mit der Dateiendung „Lua“. Diese Skripte werden dann im **BlueprintEditor** jeweils angegeben. In diesen Skripten befindet sich dann auch der Verweis („require...“ ) auf das entsprechende Modul. Der Modulaufruf darf nicht geändert werden.

## 3. Vorgaben für 3D-Modelle

Damit die Signale mit den Skripten funktionieren, müssen verschiedene Bedingungen erfüllt sein:

- Die Namen der Licht-Nodes im 3D-Modell müssen mit denen im Skript übereinstimmen
- Die Child-Namen der Signalschirme, die im BlueprintEditor angegeben werden, müssen mit denen im Skript übereinstimmen
- Für jede im BlueprintEditor zusammengebaute Kombination aus Signalschirmen und Zusatzanzeigern, muss ein passender Signalskript erstellt und im BlueprintEditor angegeben werden.

### 3.1. Allgemeines

Um für dieses und andere Signalsysteme möglichst wenig unterschiedliche 3D-Modelle erstellen zu müssen, sollten die Bezeichnungen der Licht-Nodes einem System folgen. Einmal vorgegeben, dürfen diese auch nicht mehr geändert werden.

Die in den Skripten stehenden Verweise auf die Module dürfen nicht geändert werden und passen für das entsprechende Signal oder Zusatzanzeiger bei Einhaltung aller Vorgaben.

#### 3.1.1. Regeln für die Benennung der Licht-Nodes

Die Namen der Licht-Nodes werden vorgegeben, da diese Namen im Skript festgelegt sind.

Node-Name	Bezeichnung
Rot_1	Hauptrot
Rot_3	2. Rot (nur Sperrsignale)
Gruen_1	Oben
Gelb_1	Oben
Weiss_2	links Mitte unten (Sh1)
Weiss_3	Mitte (Sh1 und Zs1 wenn kein Zs7 und kein Weiss_63 vorhanden)
Weiss_9	Kennlicht / Zusatzlicht (reduziert bzw. Wiederholer)
Weiss_63	Ersatzsignal (Mitte unterhalb von Rot oder links im Schnittpunkt vom Sh1)
Gelb_63	Vorsichtssignal (3 Signaloptyken gemeinsam)

In späteren Abschnitten folgt eine weitere Tabelle mit der Berechnung der notwendigen Code-Summe um dem Skript mitzuteilen, welche Licht-Nodes im Signal verbaut wurden.

### 3.1.2. Anmerkungen zu den Child-Namen

Für den 3D-Bau der Signalschirme werden außer den festgelegten Licht-Node-Namen keine weiteren Vorgaben benötigt. Erst die Verwendung des **BluePrintEditors** erfordert einige Einstellungen, von denen zum Beispiel die Child-Namen ganz wichtig für die Funktion der Signale sind. Diese Child-Namen sind entsprechend des Signaltyps unterschiedlich.

In den nachfolgenden Tabellen werden die jeweils zu verwendenden Child-Namen in der ersten Spalte aufgeführt. Bei diesen Child-Namen muss unbedingt auf Groß- und Kleinschreibung geachtet werden.

### 3.1.3. HP\_CODE und VR\_CODE

Die Vergangenheit hat gezeigt, dass es eine Unmenge an Sonderbauformen bei den Signalschirmen gibt. Gerade die Verwendung dieser Sonderbauformen macht eine Strecke so interessant. Leider war der Aufwand bei der Skriptgestaltung durch diese Sonderbauformen bisher sehr hoch.

Aus diesem Grunde habe ich für diese Skripte einen ganz neuen Lösungsansatz entwickelt:

- Es werden einheitliche Child-Namen auch bei unterschiedlichen Signalschirmen verwendet.
- Jede Signaloptik erhält einen eindeutigen Node-Namen, der auch bei unterschiedlichen Signalschirmen entsprechend der Funktion gleich lautet.
- Jedem dieser Node-Namen wird ein eindeutiger Zahlencode zugeordnet.
- Im Skript wird dann nur noch die Summe aller Zahlencodes je Signalschirm übermittelt und daraus alle gültigen Signaloptiken errechnet.
- Hierfür wurde im Skriptpaket eine Excel-Tabelle zur automatischen Berechnung der Code-Summe mitgeliefert. Auswählen, die sich widersprechen erzeugen eine rote Einfärbung.

Wer kein Excel zur Verfügung hat muss sich diese Code-Summe anhand der Tabelle manuell errechnen.

<b>Berechnungstabelle für KS-Signale</b>					
<b>Node-Name</b>	<b>Bezeichnung</b>	<b>Vorsignale</b>	<b>Hauptsignale</b>	<b>Sperrsignale</b>	<b>Basis-Code</b>
Rot_1	rotes Signallicht		x	x	2
Rot_3	Sperr-Rot (dual bei SH)			x	8
Gruen_1	grünes Signallicht	x	x		16
Gelb_1	gelbes Signallicht	x	x (Mehrabschnitt)		32
Weiss_1	oben	x (reduziert)			256
Weiss_2	unten / links Mitte	x (Wiederholer)	x	x	512
Weiss_3	Mitte		x	x	1024
Weiss_42	Sh1 komplett			x	2048
Weiss_9	Kennlicht / Reduziert		x		4096
Weiss_63	Ersatzsignal		x		8192
Gelb_63	Vorsichtssignal (V)		x		16384
	Vorsignalwiederholer	x			131072
	Reduzierter Abstand	x	x		262144
Code-Summe berechnet:					

**Node-Namen und Basis-Codes, die in anderen Signalsystemen möglich sind, aber in dieser Tabelle nicht aufgeführt wurden, dürfen nicht verwendet werden!**

### 3.1.4. Zusatzanzeiger

An einem Haupt- oder Mehrabschnittsignal bzw. Vorsignal können verschiedene Geschwindigkeitsanzeiger angefügt werden. Dies sind dann separate 3D-Modelle, die in der Signal-BIN über den Child-Namen angesprochen werden.

Alle möglichen Zusatzanzeiger sind im Skript als Kommentarzeile bereits aufgeführt und können durch entfernen der Kommentarstriche aktiviert werden. In den Skripten, in denen keine Zusatzanzeiger als aktivierbare Variable aufgeführt wurden, sind diese auch nicht möglich. In diesem Fall sind die Zusatzanzeiger als separates Signalobjekt zu erstellen.

### 3.2. Vorseignale

Bei allen Vorseignalen muss im Skript in der Variable VR\_CODE der Wert für die Code-Summe der verwendeten Signaloptiken angegeben werden. Weiterhin muss in der Variable VR\_SIGNAL\_HEAD\_NAME der Child-Name des Vorseignalschirmes angegeben werden.

#### 3.2.1. Vorseignal ohne Zusatzlicht

Beschreibung	Child-Name
Vorseignal ohne Zusatzlicht	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gruen_1	grünes Signallicht	Ks1	16
Gelb_1	gelbes Signallicht	Ks2	32
Code-Summe:			48

#### 3.2.2. Vorseignal ohne Zusatzlicht mit Geschwindigkeitsvoranzeiger

Beschreibung	Child-Name
Vorseignal mit Zs3	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gruen_1	grünes Signallicht	Ks1, Ks1_Slow	16
Gelb_1	gelbes Signallicht	Ks2	32
Im Skript: ZS3V_SIGNAL_HEAD_NAME = „ZS3v Signal Head“			
Code-Summe:			48

### 3.2.3. Vorsignal im verkürzten Bremswegabstand

Beschreibung	Child-Name
Vorsignal mit Zusatzlicht	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gruen_1	grünes Signallicht	Ks1	16
Gelb_1	gelbes Signallicht	Ks2	32
Weiss_1	weißes Zusatzlicht	Ks2, Ks1, deaktiviert	256
	Reduzierter Abstand		262144
Code-Summe:			262448

### 3.2.4. Vorsignalwiederholer

Beschreibung	Child-Name
Vorsignal mit Zusatzlicht	VS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	VR_CODE
Gruen_1	grünes Signallicht	Ks1, Ks1_Slow	16
Gelb_1	gelbes Signallicht	Ks2	32
Weiss_2	weißes Zusatzlicht	Ks2, Ks1, Ks1_Slow, deaktiviert	512
	Vorsignalwiederholer		131072
Code-Summe:			131632

### 3.3. Hauptsignale (einige Beispiele)

Auf Grund der Vielzahl an Möglichkeiten folgen nur einige Beispiele für die Hauptsignale. Es folgt nochmals der Hinweis, dass der Child-Name bei den Hauptsignalschirmen immer gleich ist und „HS“ lautet. Lediglich die Code-Summe wird für unterschiedliche Kombinationen an verwendeten Signaloptiken geändert.

Bei allen Haupt- und Mehrabschnittsignalen muss im Skript in der Variable HP\_CODE der Wert für die Code-Summe der verwendeten Signaloptiken angegeben werden. Weiterhin muss in der Variable HP\_SIGNAL\_HEAD\_NAME der Child-Name des Hauptsignalschirmes angegeben werden.

Zusätzlich muss bei Mehrabschnittsignalen noch die Variable VR\_SIGNAL\_HEAD\_NAME = "" aktiviert werden. Ein spezieller Wert muss nicht angegeben werden.

#### 3.3.1. Hauptsignal

Beschreibung	Child-Name
Hauptsignal mit Ersatzsignal ohne Kennlicht, ohne Sh1	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	rotes Signallicht	Hp0, Zs1, Zs7	2
Gruen_1	grünes Signallicht	Ks1	16
Weiss_63	ein weißes Signallicht	Ersatzsignal	8192
Code-Summe:			8210

Alternativ kann auch folgende Kombination verwendet werden:

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	rotes Signallicht	Hp0, Zs1, Zs7	2
Gruen_1	grünes Signallicht	Ks1	16
Weiss_3	Mitte	Ersatzsignal	1024
Code-Summe:			1042

Beschreibung	Child-Name
Hauptsignal mit Sh1	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	rotes Signallicht	Hp0, Sh1, Zs1	2
Gruen_1	grünes Signallicht	KS1	16
Weiss_2	links Mitte/unten	Sh1	512
Weiss_3	Mitte	Sh1, Zs1	1024
Code-Summe:			1554

### 3.3.2. Hauptsignal komplett mit Zs3

Beschreibung	Child-Name
Hauptsignal mit Ersatzsignal mit Kennlicht und Sh1	HS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	rotes Signallicht	Hp0, Sh1, Zs1	2
Gruen_1	grünes Signallicht	KS1	16
Weiss_2	links Mitte/unten	Sh1	512
Weiss_3	Mitte	Sh1, Zs1	1024
Weiss_9	weißes Kennlicht	Leuchtet wenn Signal deaktiviert wurde	4096
Im Skript: ZS3_SIGNAL_HEAD_NAME = „ZS3 Signal Head“			
Code-Summe:			5650

### 3.3.3. Mehrabschnittsignal

Beschreibung	Child-Name
Mehrabschnittsignal mit Ersatzsignal ohne Kennlicht, mit Sh1	MS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	rotes Signallicht	Hp0, Zs1	2
Gruen_1	grünes Signallicht	Ks1	16
Gelb_1	gelbes Signallicht	Ks2	128
Weiss_2	links Mitte/unten	Sh1	512
Weiss_3	Mitte	Sh1, Zs1	1024
Code-Summe:			1682

Beschreibung	Child-Name
Mehrabschnittsignal mit Vorsichtssignal und Sh1 ohne Kennlicht	MS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	rotes Signallicht	Hp0, Zs7	2
Gruen_1	grünes Signallicht	Ks1	16
Gelb_1	gelbes Signallicht	Ks2	128
Weiss_2	links Mitte/unten	Sh1	512
Weiss_3	Mitte	Sh1	1024
Gelb_63	3 gelbe Signallichter in V-Form	Zs7	16384
Code-Summe:			18066

### 3.3.4. Mehrabschnittsignal komplett mit Zs3 und Zs3v

Beschreibung	Child-Name
Mehrabschnittsignal mit Vorsichtssignal, Sh1 sowie Zusatzanzeiger	MS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	rotes Signallicht	Hp0, Sh1, Zs7	2
Gruen_1	grünes Signallicht	Ks1	16
Gelb_1	gelbes Signallicht	Ks2	128
Weiss_2	links Mitte/unten	Sh1	512
Weiss_3	Mitte	Sh1	1024
Weiss_9	weißes Kennlicht	Leuchtet wenn Signal deaktiviert	4096
Gelb_63	3 gelbe Signallichter in V-Form	Zs7	16384
Im Skript: ZS3_SIGNAL_HEAD_NAME = „ZS3 Signal Head“ ZS3v_SIGNAL_HEAD_NAME = „ZS3v Signal Head“			
Code-Summe:			22162

Werden feste Zusatzanzeiger verwendet, so müssen die Variablen der Zusatzanzeiger angepasst werden:

Im Skript: ZS3_SIGNAL_HEAD_NAME = „ZS3 Form Head“ ZS3v_SIGNAL_HEAD_NAME = „ZS3v Form Head“	
---	--

Die Zusatzanzeiger können in beliebigen Kombinationen mit Aktivierung oder Deaktivierung verwendet werden.

Beschreibung	Child-Name
Mehrabschnittsignal mit zusätzlichem Zs1, Sh1 sowie Zusatzanzeiger und Reduziert	MS

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_1	rotes Signallicht	Hp0, Sh1, Zs1	2
Gruen_1	grünes Signallicht	Ks1	16
Gelb_1	gelbes Signallicht	Ks2	128
Weiss_2	links Mitte/unten	Sh1	512
Weiss_3	Mitte	Sh1	1024
Weiss_9	weißes Kennlicht	Leuchtet wenn Signal deaktiviert oder bei verkürztem Abstand (Rd)	4096
Weiss_63	weißes Signallicht links Mitte	Zs1	8192
	Code für Reduzierten Abstand		262144
Im Skript: ZS3_SIGNAL_HEAD_NAME = „ZS3 Signal Head“ ZS3v_SIGNAL_HEAD_NAME = „ZS3v Signal Head“			
Code-Summe:			276114

### 3.3.5. Ausblenden von Signaloptiken

Um weniger unterschiedliche Signalschirme bauen zu müssen, ist es mit diesen Skripten möglich, vorhandene Signallaternen auszublenden und die vorhandenen Licht-Nodes dauerhaft dunkel zu schalten.

Damit Signaloptiken ausgeblendet werden können, müssen 2 Bedingungen erfüllt werden:

1. Das 3D Modell wurde so gebaut, dass eine komplette Signallaterne über einen Node-Namen wie eine Signaloptik angesprochen werden kann
2. Der Skript muss beim Laden der Strecke genau diesen Node-Namen ansprechen und ähnlich wie ein Signallicht dunkel geschaltet wird, die Signaloptik ausblenden.

Um diese Funktionalität möglichst einfach zu gestalten, wird eine zusätzliche Code-Variable (OBJ\_CODE) eingeführt, die die Summe aller Signaloptik-Codes enthält die ausgeblendet werden sollen. Da die Signaloptik dann existiert, muss die Signaloptik auch im HP\_CODE mit eingerechnet werden. Die Node-Namen der Signaloptiken sind zu den Licht-Nodes analog und werden nur die Bezeichnung OBJ\_ vorangestellt. Soll nun zum Beispiel die untere Signaloptik vom Rangiersignal ausblendbar gebaut werden, gilt folgendes:

Im Signalschirm vorhandene Signaloptiken: Rot\_1, Gruen\_1, Weiss\_2, Weiss\_3

Name der Licht-Node: Weiss\_2  
 Name der Signallaterne: OBJ\_Weiss\_2  
 Code im Skript: HP\_CODE = 1554  
 OBJ\_CODE = 512

Da nun Weiss\_2 ausgeblendet wurde, wird nun auch kein Sh1 mehr signalisiert. Lediglich das Zs1 ist über Weiss\_3 möglich.

### 3.4. Sperrsignale

Bei den Sperrsignalen können wahlweise Rot\_1 und / oder Rot\_3 verwendet werden.

#### 3.4.1. Sperrsignal ohne Deckungssignalfunktion

Beschreibung	Child-Name
Sperrsignal	SH

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_3	1 oder 2 rote Signallichter	Hp0	8
Weiss_42	zwei weiße Signallichter	Sh1	2048
Code-Summe:			2056

#### 3.4.2. Sperrsignal und Deckungssignal

Beschreibung	Child-Name
Sperrsignal, auch als Deckungssignal einsetzbar	SP

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Rot_3	1 oder 2 rote Signallichter	Hp0	8
Weiss_2	linkes weißes Signallicht	Sh1	512
Weiss_3	rechtes weißes Signallicht	Sh1, Leuchtet bei Funktion als Deckungssignal	1024
Code-Summe:			1544

### 3.4.3. Wartesignal

Beschreibung	Child-Name
Rangiersignal Ra12 (dunkel bei Zugfahrt wenn zwischen Hauptsignallinks eingebettet)	Ra12

Node-Name	Beschreibung	Leuchtet bei Signalbegriff	HP_CODE
Weiss_42	zwei weiße Signallichter	Ra12	2048
Code-Summe:			2048

### 3.5. Zusatzsignale und Zusatzanzeiger

Sämtliche Zusatzsignale und Zusatzanzeiger schalten ausschließlich Texturen. Deshalb gibt es hier keine Licht-Nodes, sondern es muss lediglich auf die richtige Verwendung des Child-Namens geachtet werden.

Alle folgenden Zusatzanzeiger sind separate Signalobjekte als eigenes BluePrint und mit eigenen Links, die später in vorgesehener Weise in der Nähe der Hauptsignale platziert werden können.

Es gibt immer 2 Varianten. Eine normale mit Basisgeometrie und einem Child und eine Variante ohne Child nur mit einer Basisgeometrie.

#### 3.5.1. Fahrtanzeiger

Child-Name	Beschreibung
FAZ Signal Head	Fahrtanzeiger
FAZ	Fahrtanzeiger (ohne Child im BluePrint)

Der Fahrtanzeiger besitzt in der einseitigen Ausführung nur einen Link und spricht über den Buchstaben „V“ die Textur an. Werden beim Fahrtanzeiger in der Signal-BIN 2 Links festgelegt, so wird für die Gegenrichtung die Textur über dem Buchstaben „R“ aktiviert.

Die Buchstaben sind im Skript festgelegt und müssen verwendet werden.



#### 3.5.2. Abfahrtsignal

Child-Name	Beschreibung
ZP9 Signal Head	Fahrtanzeiger für DR und DB
ZP9	Fahrtanzeiger für DR und DB (ohne Child im BluePrint)

Die Textur für den Abfahrauftrag wird aus dem Signalskript in der Variable SIGNAL\_TYP mit übergeben und kann frei gewählt werden.



### 3.5.3. Richtungsanzeiger

Child-Name	Beschreibung
ZS2v Signal Head	Licht - Richtungsvoranzeiger Zs2v
ZS2v	Licht - Richtungsvoranzeiger Zs2v (ohne Child im BluePrint)
ZS2 Signal Head	Licht - Richtungsanzeiger Zs2
ZS2	Licht - Richtungsanzeiger Zs2 (ohne Child im BluePrint)

Die Texturen werden über die Eintragungen im Bereich PrimaryNamedTextureSet geladen.  
Die Buchstaben für die Texturen können frei gewählt werden.

Bei dem Zs2 muss grundsätzlich zwischen der Varianten mit einem Link und mehreren Links unterschieden werden:

- Besitzt das Zs2 nur einen Link, so wird der Skript „DEs KS Zs2.lua“ verwendet. Hierbei werden die Buchstaben bei den Links im Hauptsignal eingetragen. Und der Link 0 des Zs2 wird direkt vor den Link 0 des Hauptsignallink 0 gesetzt.
- Besitzt das Zs2 mehrere Links, so werden die Links 1+ als Ziellinks verwendet. In diesem Falle muss der Skript „DEs KS Zs2 T.lua“ verwendet werden. Die Buchstaben für die Richtungen werden nun direkt im Zs2 T eingetragen.

### 3.5.4. Gleiswechselanzeiger

Child-Name	Beschreibung
ZS6 Signal Head	Gleiswechselanzeiger Zs6
ZS6	Gleiswechselanzeiger Zs6 (ohne Child im BluePrint)

Die Texturen werden über die Eintragungen im Bereich PrimaryNamedTextureSet geladen.  
Die Buchstaben für die Texturen können frei gewählt werden.

Das Zs6 besitzt grundsätzlich mindestens 2 Links. Es können auch Varianten mit mehr als 2 Links erstellt werden. Die Links 1+ werden immer als Ziellinks verwendet. Der Skript „DEs HL Zs6.lua“ wird verwendet. Die Buchstaben für die Richtungen sind nicht vorgegeben und werden später direkt im Zs6 eingetragen.

### 3.5.5. Geschwindigkeitsanzeiger

Child-Name	Beschreibung
Zs3v Signal Head	Licht - Geschwindigkeitsvoranzeiger Zs3v
Zs3v	Licht - Geschwindigkeitsvoranzeiger Zs3v (ohne Child im BluePrint)
Zs3 Signal Head	Licht - Geschwindigkeitsanzeiger Zs3
Zs3	Licht - Geschwindigkeitsanzeiger Zs3 (ohne Child im BluePrint)
Zs3v Form Head	Form - Geschwindigkeitsvoranzeiger Zs3v
Zs3v	Form - Geschwindigkeitsvoranzeiger Zs3v (ohne Child im BluePrint)
Zs3 Form Head	Form - Geschwindigkeitsanzeiger Zs3
Zs3	Form - Geschwindigkeitsanzeiger Zs3 (ohne Child im BluePrint)

Die Texturen werden über die Eintragungen im Bereich PrimaryNamedTextureSet geladen.

Geschwindigkeitsanzeiger und Geschwindigkeitsvoranzeiger können in ein Signal-Blueprint von einem Hauptsignal integriert werden. Hierzu muss im Blueprint das Child mit eingetragen und im Skript die Variable des Child-Namens aktiviert werden.

Form – Geschwindigkeitsanzeiger bzw. Form – Geschwindigkeitsvoranzeiger besitzen in der Regel keinen separaten Kopf (Variante ohne Child im Blueprint). Die Texturen für die Zahlen werden dennoch im Bereich PrimaryNamedTextureSet vereinbart.

### 3.6. Signale ohne Signalschirm

Um erweiterte Funktionen der Signale zu realisieren, sind zusätzliche Signale ohne Signalschirm notwendig. Hierbei wird als 3D Objekt lediglich ein Kasten benötigt um das Objekt packen zu können.

Ausführungen:

- Dummy
- Hauptsignal ohne Mast
- LZB-Teilblock

Die Funktionalitäten für diese Objekte sind in meinen regulären Signalsystemen (z.B. HV-Signalsystem) hinreichend erläutert. Für die vollständige Funktion eines Signalsystems werden diese besonderen Signale jedoch auch hier benötigt.

Bezeichnung	HP_SIGNAL_HEAD_NAME	VR_SIGNAL_HEAD_NAME	Stopping	Anzahl Links
DEs KS HS Dummy	Dummy		eTrue	2 bis beliebig
DEs KS HS LZB	LZB		eTrue	1
DEs KS HS OhneMast	OSS	„	eTrue	2 bis beliebig

Bezeichnung	VR_SIGNAL_HEAD_NAME	Stopping	Anzahl Links
DEs KS VS Dummy	Dummy	eFalse	2
DEs KS VS rh Dummy	Dummy	eFalse	2
DEs KS VS Wh Dummy	Dummy	eFalse	2

① **Bei Anbausignalen wird das Stopping immer auf „eFalse“ gesetzt.**

### 4. Skripte

Für jede Kombination aus Signalschirmen und Zusatzanzeigern muss genau ein Skript im Verzeichnis Skripte unter Ihrem Source-Verzeichnis existieren.

Es wurden ja nur Beispielskombinationen mitgeliefert. Nun liegt es an Ihnen, nach persönlichem Geschmack Skriptnamen festzulegen. Durch die Verwendung des Blueprint-Editors wird beim Export ein gleichnamiger Skript mit der Dateierweiterung „.out“ im entsprechenden Ordner unter Assets erstellt. Dieser Skript benötigt dann die Module und die Optionsdatei im Unterordner bei Schuster/Freeware, damit das Signal funktionieren kann.

Es können also auch die mitgelieferten Skriptnamen geändert werden. Lediglich der Inhalt der Skripte folgt den speziellen Vorgaben, die weiter oben genannt wurden. So darf auch das unter „require“ zugeordnete Modul inklusive der Pfadangabe keinesfalls geändert werden.

Im Skript wird je nach Signalschirm der errechnete HP\_CODE und / oder der VR\_CODE hinter der entsprechenden Variablen angegeben.

Auf den nächsten Seiten folgen Tabellen mit den im Skriptpaket vorhandenen Skripten.

### Bemerkungen:

- Kasten = Es wird ein 3D-Objekt ohne Licht-Nodes in Form eines Gehäuses benötigt
- Anbau = Spezielles Signal zur Realisierung eines Hauptsignals an einer DKW

Signaltyp	Skript-Name	Child-Name				Bemerkung
		Haupt-signal	Vor-signal	Zusatzanzeiger HS	Zusatzanzeiger VS	
<b>Vorsignale</b>						
Vorsignal	DEs KS VS.lua		VS			
Vorsignal im reduzierten Abstand	DEs KS VS rd.lua		VS			
Vorsignal als Wiederholer	DEs KS VS Wh.lua		VS			
<b>Hauptsignale</b>						
Hauptsignal	DEs KS HS.lua	HS				
Mehrabschnittsignal	DEs KS HS VS.lua	MS	MS			
Mehrabschnittsignal im reduzierten Abstand	DEs KS HS VS rd.lua	MS	MS			
<b>Sperrsignale</b>						
Sperrsignal	DEs KS SH.lua	SH				
Sperrsignal als Deckungssignal	DEs KS SP.lua	SP				
Wartesignal	DEs KS Warte.lua	Ra12				
<b>Sonderbauformen</b>						
Hauptsignal – Dummy	DEs KS HS Dummy.lua	Dummy				Kasten
Hauptsignal als Anbauvariante (speziell für DKW-Bereiche)	DEs KS Anbau HS.lua	HS				Anbau
Weitere Bauformen siehe Punkt 3.6.						

Signaltyp	Skript-Name	Child-Name				Bemerkung
		Haupt-signal	Vor-signal	Zusatzanzeiger HS	Zusatzanzeiger VS	
<b>Zusatzsignale</b>						
Fahrtanzeiger (einfach oder doppelt)	DEs KS FAZ.lua			FAZ Signal Head		
Fahrtanzeiger (einfach oder doppelt)	DEs KS FAZ.lua			FAZ		ohne Child
Abfahrtsignal - Deutsche Bundesbahn	DEs KS Zp9 DB.lua			ZP9 Signal Head		
Abfahrtsignal - Deutsche Bundesbahn	DEs KS Zp9 DB.lua			ZP9		ohne Child
Abfahrtsignal - Deutsche Reichsbahn	DEs KS Zp9 DR.lua			ZP9 Signal Head		
Abfahrtsignal - Deutsche Reichsbahn	DEs KS Zp9 DR.lua			ZP9		ohne Child
Richtungsanzeiger für Zielgleise (ab 1T = 2 Links)	DEs KS Zs2 T.lua			ZS2 Signal Head		
Richtungsanzeiger für Zielgleise (ab 1T = 2 Links)	DEs KS Zs2 T.lua			ZS2		ohne Child
Richtungsanzeiger	DEs KS Zs2.lua			ZS2 Signal Head		
Richtungsanzeiger	DEs KS Zs2.lua			ZS2		ohne Child
Richtungsvoranzeiger	DEs KS Zs2v.lua				ZS2v Signal Head	
Richtungsvoranzeiger	DEs KS Zs2v.lua				ZS2v	ohne Child
Kurze Fahrt Trigger	DEs KS Zs3 KF Tr.lua					Kasten
Geschwindigkeitsanzeiger für Kurze Fahrt	DEs KS Zs3 KF.lua			ZS3 Signal Head		
Geschwindigkeitsanzeiger	DEs KS Zs3.lua			ZS3 Signal Head		
Geschwindigkeitsanzeiger	DEs KS Zs3.lua			ZS3		ohne Child
Geschwindigkeitsanzeiger fest	DEs KS Zs3Fo.lua			ZS3		ohne Child
Geschwindigkeitsvoranzeiger	DEs KS Zs3v.lua				ZS3v Signal Head	
Geschwindigkeitsvoranzeiger	DEs KS Zs3v.lua				ZS3v	ohne Child
Geschwindigkeitsvoranzeiger fest	DEs KS Zs3vFo.lua				ZS3v	ohne Child
Gleiswechselanzeiger (ab 1T = 2 Links)	DEs KS Zs6.lua			ZS6 Signal Head		
Gleiswechselanzeiger (ab 1T = 2 Links)	DEs KS Zs6.lua			ZS6		ohne Child

## 5. BluePrintEditor

Um das 3D-Modell mit dem Skript zu verbinden, ist der BluePrintEditor erforderlich. Hier werden einige Einstellungen für das Signal vorgenommen. Aus den erstellten Signalschirmen und Zusatzanzeigern kann nun unter Auswahl des richtigen Signalskriptes ein funktionierendes Signal erstellt werden.

Dieses Thema kann nur angerissen und nicht erschöpfend beschrieben werden. Deshalb gibt es hier nur einige Hinweise, die bei speziellen Bauarten von Signalschirmen auch noch abweichen können.

### 5.1. Grundsätzliches

Kenntnisse zur Benutzung des BluePrintEditors setze ich voraus. Ich werde nur auf die relevanten Einstellungen für die Funktion des 3D-Modells mit dem Skript eingehen.

### 5.2. Köpfe

**Für alle Signalschirme und Zusatzanzeiger muss vorab ein AnimSceneryBluePrint erstellt werden.**

**Später werden diese Signalschirme und Zusatzanzeiger in einem SignalBluePrint zusammengefügt.**

Die 3D-Modelle der Signalschirme bzw. Zusatzanzeiger-Objekte werden zuerst als ein separater „Kopf“ im BluePrintEditor definiert. Hierfür wird in der Regel das AnimSceneryBluePrint verwendet.

Im BluePrint wird im Bereich BrowseInformation unbedingt die **Category „eExcludeFromBrowserList“** angegeben, damit der Kopf später im Editor nicht auswählbar ist, sondern verborgen bleibt.

Ein Name muss bei allen BluePrints angegeben werden. Jedoch DisplayName und Beschreibung können entfallen.

Ein weiterer wichtiger Eintrag ist die **Geometrie** im Bereich **RenderComponent**, in der der Signalschirm bzw. der Zusatzanzeiger angegeben wird.

Lediglich bei den Form-Geschwindigkeitsanzeigern gibt es keinen separaten Kopf, damit zusätzlich Zahlen (im ID-Feld eingetragen) direkt im Schild angezeigt werden können.

### 5.3. Einstellungen für Signale und Zusatzanzeiger

Zuerst wird ein neues BluePrint geöffnet. Wir wählen ein „**Signal BluePrint**“ und nehmen sorgfältig alle notwendigen Eintragungen vor:

- ① **Name**
  - **Name**

Ein eindeutiger Name muss bei allen BluePrints angegeben werden. Dieser wird in der Tracks.bin vermerkt und erscheint auch bei Fehlermeldungen.

- ① **BrowseInformation**
  - **DisplayName**
  - **Description**

In dem Bereich DisplayName ist grundsätzlich die Sprache English und German mit einer passenden Bezeichnung auszufüllen. Die restlichen Sprachen werden wohl auf English verwiesen, wenn nichts ausgefüllt ist. Man sollte sich vorher ein gutes System überlegen, um die Signale später eindeutig identifizieren zu können.

Keinesfalls darf sich die Bezeichnung mit anderen Editorobjekten überschneiden, da bei gleichzeitiger Freischaltung ungewollte Ergebnisse entstehen.

Auch ein Eintrag im Feld Description kann später bei der Auswahl im Welteditor hilfreich sein, da diese Information als Tool-Tipp unterhalb des Mauszeigers angezeigt wird.

### ① **BrowseInformation**

- **Category**

Der Eintrag Category legt fest, in welcher Gruppe später im Welteditor das Signal gelistet werden soll. Wir verwenden für alle Signale die Auswahl: eTrackInfrastrukture

### ① **\_object**

- **PrimaryNamedTextureSet**
- **SecondaryTextureSet**

Im Bereich PrimaryNamedTextureSet und SecondaryTextureSet werden die Textursets für die erste und zweite Zeile der Mastnummer, soweit vorhanden, angegeben. Ist keine Mastnummer vorhanden, können diese Angaben entfallen. Auch wenn diese Bereiche frei bleiben, können später die im ID-Feld eingetragenen Bezeichnungen zum Debug verwendet werden.

Bei allen **Zusatzanzeigern** wird im PrimaryNamedTextureSet das zu verwendende Texturset für die darzustellenden Zeichen angegeben.

### ① **\_object**

- **GeometryID**

Im Eintrag „GeometryID“ wird die Mastgeometrie angegeben. Hierbei handelt es sich **nicht** um die Signalschirme. Diese werden später im Bereich \_container als Children definiert.

### ① **\_signal**

- **NumberOfTrackLinks**

Im Bereich **\_signal** werden nun weitere Signaleigenschaften definiert. Dort findet sich als erster Eintrag NumberOfTrackLinks, welcher die Anzahl der Ziellinks zuzüglich des Link 0 beinhaltet. Für ein Blocksignal (OT), alle Vorsignale und bei allen Zusatzanzeigern bis auf die „T“-Varianten für Zs2 und Zs6 wird hier der Wert 1 eingetragen. Bei allen Hauptsignalen, die Fahrstraßen über Weichen haben, zählen wir für jedes Ziel den Wert 1 hinzu. Liegt hinter dem Hauptsignal zum Beispiel eine Weiche, dann wird der Wert 3 in das Feld NumberOfTrackLinks eingetragen.

### ① **\_signal**

- **Stopping**

In das Kästchen bei Stopping wird bei allen Hauptsignalen und Sperrsignalen ein Häkchen gesetzt. Hierdurch wird später ein neuer Blockabschnitt, beginnend am Link 0, gebildet.

Dies ist die einzige Eigenschaft, die KI-Züge vom Signal kennen und darauf auch reagieren.

### ① **\_signal**

- **JunctionSignalType**

Bei der Auswahlliste JunctionSignalType sollte bei allen Signalen der Wert **eJunctionTypSpeed** ausgewählt werden, um später im Signal-Flyout die Eingabemöglichkeit der Richtungsangabe und der Geschwindigkeit zu haben.

### ① **\_signal**

- **ControlMode**

Die Auswahl im Bereich ControlMode ist für die normale Funktion weniger von Bedeutung. Wird jedoch beabsichtigt, das Signalsystem gemeinsam mit der von mir erstellen 2DMapPro zu nutzen, dann sollte folgende Einstellung vorgenommen werden:

Hauptsignale ohne Vorsignalfunktion:	eControlModeControlled
Mehrabschnittsignale:	eControlModeControlledCallOn
Alle anderen Signale:	eControlModeAutomatic

Nur mit diesen Einstellungen werden dann die passenden Mastsymbole in der 2DMapPro angezeigt.

## ① **\_script**

Nun kommt im Bereich `_script` die wichtige Entscheidung, welcher Skript zu verwenden ist. Wie schon weiter oben geschrieben, gibt es in der Regel für eine bestimmte Kombination aus Signalschirmen immer nur einen einzigen Skript, der zu verwenden ist. Schauen Sie also genau in der unter Punkt 4. aufgeführten Tabelle nach, welcher Skript zu ihrem Signal passt. Ist dieser nicht vorhanden, dann kopieren Sie am besten einen vorhandenen Skript und passen dessen Einstellungen durch Aktivierung von Variablen an und tragen den jeweils gültigen CODE für den oder die Signalschirme ein.

Der dort einzutragende Skript wird mit dem Verzeichnisnamen oberhalb von Provider- Produktnamen und ohne Dateiendung angegeben.

Ändern Sie keinesfalls den Verweis auf das dazugehörige Modul!

Beispiel für ein einfaches Hauptsignal:      `RailNetwork\Signals\German KS\Skripte\DEs KS HS`

Der Skript muss als Datei im Verzeichnis Source existieren, sonst schlägt der Export fehl. Im oben genannten Beispiel also genau diese Datei:

```
Source\[Provider]\[Produkt]\RailNetwork\Signals\German KS\Skripte\DEs KS HS.lua
```

Die Werte für `[Provider]` und `[Produkt]` entsprechen Ihren Verzeichnisnamen.

## ① **\_container**

- **Children**

Erst jetzt, im Bereich `_container` -> `Children` werden die Einstellungen, die die einzelnen Signalschirme und Zusatzanzeiger betreffen, festgelegt.

Für jeden Signalschirm bzw. Zusatzanzeiger muss ein Child hinzugefügt werden (Add Element...).

Der Übersicht halber beginnen wir immer mit dem Hauptsignal. Als 2. Child dann das Vorsignal (wenn vorhanden) und zum Schluss die Zusatzanzeiger (wenn vorhanden).

Im Feld `ChildName` wird der `ChildName` aus der Tabelle (Punkt 3.3) für den entsprechenden Signalschirm eingetragen. Hierbei ist auf die korrekte Schreibweise zu achten.

Es kann immer nur ein `Zs3` und/oder `Zs3v` Zusatzanzeiger hinzugefügt werden. Die gleichzeitige Verwendung von **Licht- und Form-** Geschwindigkeits- bzw. voranzeigern ist nicht möglich.

## ① **\_container**

- **Children**
  - **BlueprintID**

An dieser Stelle wird die XML-Datei vom Signalschirm eingetragen.

Die Einstellungen hierzu sind bei den Signalschirmen sehr unterschiedlich und werden deshalb nicht näher aufgeführt. Anzumerken ist jedoch, dass bei den Signalschirmen die `Category: eExcludeFromBrowserList` auszuwählen ist und in diesem BluePrint kein Skript angegeben wird.

## ① **\_container**

- **Children**
  - **Matrix**

Die Matrix ermöglicht es, den Signalschirm am Mast entsprechend zu positionieren.

#### **5.4. Export des BluePrints**

Im unteren Bereich sollte man sich unbedingt den Bereich Output aufklappen [+], damit man die Meldungen verfolgen kann.

Wenn alles richtig eingegeben wurde, wird das BluePrint mit der Funktionstaste F7 nach Assets exportiert. Nur wenn am Ende des Protokolls „Export Succeeded“ steht, ist der Export erfolgreich abgeschlossen. Anderen falls muss man die Einstellungen korrigieren.

Anschließend sollte das Signal im Welteditor des Train Simulators zur Verfügung stehen.

#### **5.5. Einsatz der Signale in einer Train Simulator Strecke**

Wenn Sie das Signal in eine Strecke einbauen, muss das Signal nach dem Neustart der Strecke das richtige Signalbild, zum Beispiel Hp0 anzeigen. Leuchten auf dem Signalschirm dann noch alle Signallichter, so gibt es ein Problem, welches an falschen Einstellungen liegen kann.

##### **Prüfen Sie dann folgende Dinge:**

- Starten Sie LogMate mit dem TS und kontrollieren Sie, ob dort Fehler angezeigt werden
- Passt der Skript auch zu den verwendeten Signalschirmen, die im BluePrint verwendet wurden?
- Sind alle Child-Namen richtig geschrieben?
- Wurde die richtige Code-Summe berechnet und im Skript eingetragen?
- Prüfen Sie das 3D-Modell, ob die Licht-Node-Namen stimmen

Ich hoffe, dass ich das Thema hinreichend ausführlich und richtig beschrieben habe.

Sollte es Probleme geben, bitte ich darum, im Forum bei einem geeigneten Thema nachzufragen.

So können Fragen schnell geklärt werden und die Lösungen stehen Vielen zur Verfügung.

Danke.

Ich wünsche viel Erfolg beim Einsatz der Skripte  
Mathias Gundlach

---