

# Loksim3D Funktionsdokumentation

Lukas Haselsteiner      Uwe Klein      Martin Mory

Juli 2020

---

# 1 Vorwort

---

Herzlich Willkommen zur Funktionsdokumentation des Loksim3D. Diese Dokumentation stellt eine Beschreibung neuer Funktionen ab Version 2.8.2 dar. Diese Dokumentation dient hauptsächlich für erfahrende Loksimmer und soll insbesondere möglichst viel Informationen zum Objekt-, Führerstand- und Streckenbau bieten.

Wir hoffen, dass nach dem Studium der Basisdokumentation diese Datei eine gute Hilfeleistung für weiterführende Themen bieten kann. Wir freuen uns über jedes neue Addon für den Loksim - probieren Sie den LoksimEdit einfach selbst aus.

Der Loksim ist ein Freeware-Projekt und wird von den Projektbeteiligten ausschließlich in der Freizeit weiterentwickelt. Wenn Sie Interesse an einer Mitarbeit am Loksimprojekt haben, freuen wir uns über Ihre Nachricht. Es gibt viel zu tun, u.a.

- Programmierung neuer oder verbessern bestehender Funktionen: Für eine Mitarbeit sollten objektorientierte und prozedurale Prinzipien kein Problem sein und C++ beherrscht werden
- Bei der Dokumentation des Loksims mithelfen
- Beim Übersetzen des Programms bzw. der Dokumentation helfen
- (Mit)arbeit an einer neuen Demo-Strecke bzw. Lok
- ...

---

# Inhaltsverzeichnis

---

<b>1</b>	<b>Vorwort</b>	<b>1</b>
1.1	Version 2.10 . . . . .	7
1.1.1	Neue Funktionen . . . . .	7
1.1.2	Kleinere Änderungen . . . . .	7
1.1.3	Fehlerkorrekturen . . . . .	7
<b>I</b>	<b>Simulator</b>	<b>8</b>
<b>2</b>	<b>Steuerung</b>	<b>9</b>
2.1	Diverses . . . . .	9
2.1.1	Zuglängenzähler / Wegmessung . . . . .	9
<b>3</b>	<b>Optionen</b>	<b>10</b>
3.1	Darstellung . . . . .	10
3.2	Simulation . . . . .	10
3.3	Konfigurationsdateien . . . . .	10
3.3.1	Datenpfade . . . . .	10
3.4	Besondere Einstellungen . . . . .	11
<b>4</b>	<b>PackageManager</b>	<b>12</b>
4.1	Übersicht . . . . .	12
4.2	Funktionsweise . . . . .	12
4.3	Deinstallation während Installation . . . . .	13
<b>II</b>	<b>Editor</b>	<b>14</b>
<b>5</b>	<b>Objekte</b>	<b>15</b>
5.1	Objekte . . . . .	15
5.1.1	Objekteigenschaften . . . . .	15

5.1.2	Punkte . . . . .	16
5.1.3	Flächen . . . . .	16
5.1.4	Normalenvektoren . . . . .	16
5.1.5	Zoomen/Verschieben . . . . .	17
5.2	Gruppenobjekte . . . . .	17
5.2.1	Verschieben . . . . .	17
5.3	Externe 3D-Objektmodellformate . . . . .	18
5.3.1	Konvertieren externer 3D-Objektmodellformate	18
5.4	Gleise . . . . .	18
5.5	Textur . . . . .	19
5.5.1	Transparenz . . . . .	19
5.6	Texturnutzung optimieren . . . . .	20
5.6.1	Empfehlungen . . . . .	20
5.6.2	Einfluss der Texturen auf Performance . . . .	21
5.7	Logische Ausdrücke . . . . .	23
5.7.1	Operanden (Variablen) . . . . .	23
5.7.2	Operatoren . . . . .	25
5.7.3	Funktionen . . . . .	26
5.7.4	Beispiele . . . . .	27
5.7.5	Einfluss auf Performance . . . . .	27
5.8	Sichtbarkeitssteuerung . . . . .	28
5.8.1	Sichtbarkeitssteuerung in Gruppenobjekten .	28
5.8.2	Sichtbarkeitssteuerung Streckenobjekte . . . .	29
5.8.3	Sichtbarkeitssteuerung über Eigenschaften der Objekte . . . . .	29
5.9	Dynamische Schriften . . . . .	30
5.9.1	Syntax in EBNF . . . . .	30
5.9.2	Erklärung . . . . .	31
5.9.3	Performance . . . . .	32
5.9.4	Beispiele . . . . .	32

<b>6</b>	<b>Strecken</b>	<b>34</b>
6.1	Gleis . . . . .	34
6.1.1	Gleiseigenschaften . . . . .	34
6.2	Streckenobjekte . . . . .	35
6.2.1	Wiederholung an Achse . . . . .	35
<b>7</b>	<b>Loks</b>	<b>36</b>
7.1	Führerstand . . . . .	36
7.2	Dialog Motor (Antriebsdaten) . . . . .	36
7.2.1	maximale Geschwindigkeit (km/h) . . . . .	37
7.2.2	maximale Motorspannung (V) . . . . .	37
7.2.3	Wirkungsgrad der Motoren (%) . . . . .	37
7.2.4	maximale Lokleistung (kW) . . . . .	37
7.2.5	Maximaler Motorstrom (A) . . . . .	38
7.2.6	Wirkungsgrad Transformator (%) . . . . .	38
7.2.7	Oberspannung (V) . . . . .	38
7.2.8	maximale Zugkraft (kN) . . . . .	38
7.2.9	Motorparameter . . . . .	38
7.2.10	Standardwerte für E-Loks, Drehstromloks und Dieselloks . . . . .	40
7.2.11	Diagramm . . . . .	41
7.2.12	Bemerkung . . . . .	41
7.3	Indusi/PZB . . . . .	41
7.3.1	Art der Indusi . . . . .	41
7.3.2	Sounds . . . . .	42
7.3.3	Leuchtmelder und Anzeigen . . . . .	42
7.3.4	PZB Einstellungen . . . . .	42
7.4	Türsteuerung . . . . .	44
7.5	Diverses . . . . .	44
7.5.1	Wegmessung / Zuglängenzähler . . . . .	44
<b>8</b>	<b>Wettereditor</b>	<b>45</b>
8.1	Niederschlag . . . . .	45

<b>9</b>	<b>Allgemeines</b>	<b>46</b>
9.1	Dateien überschreiben . . . . .	46
9.2	Package erzeugen . . . . .	47
9.2.1	Diese Dateien löschen / Packages deinstallieren	48
9.2.2	l3dmeta Dateien . . . . .	48
9.3	Automatische Sicherung von Dateien . . . . .	49
9.4	Logging . . . . .	49
<b>III</b>	<b>Anhang</b>	<b>51</b>
<b>10</b>	<b>Versionshistorie</b>	<b>52</b>
10.1	Version 2.10 . . . . .	52
10.1.1	Neue Funktionen . . . . .	52
10.1.2	Kleinere Änderungen . . . . .	52
10.1.3	Fehlerkorrekturen . . . . .	52
10.2	Version 2.9.6 . . . . .	52
10.2.1	Neue Funktionen . . . . .	53
10.2.2	Kleinere Änderungen . . . . .	53
10.2.3	Fehlerkorrekturen . . . . .	53
10.3	Version 2.9.5 . . . . .	53
10.3.1	Neue Funktionen . . . . .	53
10.3.2	Kleinere Änderungen . . . . .	53
10.3.3	Fehlerkorrekturen . . . . .	53
10.4	Version 2.9.4 . . . . .	53
10.4.1	Neue Funktionen . . . . .	54
10.4.2	Kleinere Änderungen . . . . .	54
10.4.3	Fehlerkorrekturen . . . . .	54
10.5	Version 2.9.3 . . . . .	54
10.5.1	Neue Funktionen . . . . .	54
10.5.2	Kleinere Änderungen . . . . .	55
10.5.3	Fehlerkorrekturen . . . . .	55
10.6	Version 2.9.2 . . . . .	56
10.6.1	Neue Funktionen . . . . .	56

10.6.2	Kleinere Änderungen . . . . .	56
10.6.3	Fehlerkorrekturen . . . . .	57
10.7	Version 2.9.1 . . . . .	57
10.7.1	Neue Funktionen . . . . .	57
10.7.2	Kleinere Änderungen . . . . .	57
10.7.3	Fehlerkorrekturen . . . . .	58
10.8	Version 2.9 . . . . .	58
10.8.1	Neue Funktionen . . . . .	58
10.8.2	Kleinere Änderungen . . . . .	59
10.8.3	Fehlerkorrekturen . . . . .	59
10.9	Version 2.8.3 . . . . .	60
10.9.1	Neue Funktionen . . . . .	60
10.9.2	Kleinere Änderungen . . . . .	60
10.9.3	Fehlerkorrekturen . . . . .	61
10.10	Version 2.8.2a . . . . .	61
10.11	Version 2.8.2 . . . . .	61

## **1.1 Version 2.10**

### **1.1.1 Neue Funktionen**

- Loksim: RailDriver Integration
- LoksimEdit: Überarbeitung Führerstandseditor
- LoksimEdit: "Grab Geste" fuer KBS-, Strecke-, Objektbitmap- und Führerstandsbitmap-View
- LoksimEdit: Letzer Zustand laden im Create Package Dialog

### **1.1.2 Kleinere Änderungen**

- Loksim: SEP um textEBuLa erweitert
- Verbesserte Berechnung der Bounding Box von 3D-Fonts
- LoksimEdit: 2D-View aktualisiert sich wenn Objekte geändert werden
- LoksimEdit: Englische Übersetzung für (neue) Teile des LoksimEdit
- LoksimEdit: Sichtwinkel bei Update in KBS-, und Strecke-View wiederherstellen
- LoksimEdit: Im Führerstandeditor sind veraltete Optionen deaktiviert
- PackageManager: Option Transaktionen zu deaktivieren

### **1.1.3 Fehlerkorrekturen**

- Simulator: Bugfix bei Indusi zusätzlich: Korrektur der Berechnung der Signalposition
- Simulator: Bugfix Fst-Überblendung im Stand
- Simulator: Crash beim Blättern im Fahrtenschreiber
- LoksimEdit: Bugfix (Crash) Speichern von Backups wenn Pfad nicht beschreibbar



---

## I. Simulator

---

---

## 2 Steuerung

---

### 2.1 Diverses

#### 2.1.1 Zuglängenzähler / Wegmessung

ab 2.8.3

Moderne Loks sind häufig mit einem Zuglängenzähler (auch als Wegmessung bekannt) ausgerüstet. Einsatzgebiet ist beispielsweise das Ende einer Langsamfahrstelle: Sobald das Tzfz das Ende des Gefahrenbereichs erreicht, startet der Tzfz die Wegmessung. Nachdem die Lok eine Zuglänge zurückgelegt hat, ertönt ein akkustisches Signal. Dadurch weiß der Tzfz, dass es sicher ist, den Zug zu beschleunigen.

Falls ein Loksim-Führerstand mit dem Zuglängenzähler ausgestattet ist, kann dieser über die in den Optionen einstellbare Tastenkombination, gestartet werden. Sobald der Zug die Zuglänge abgefahren hat, ertönt ein Ton. Optional kann auch beim Start der Wegmessung ein akkustisches Signal erfolgen. Daneben kann die Wegmessung auch mit einem Doppelklick auf die Sifa-Taste bzw. zweimaligem Loslassen der Sifa-Taste (reale Sifa-Taste Joystickssteuerung) gestartet werden. Dieses Verhalten ist in den Optionen abschaltbar.

---

## 3 Optionen

---

### 3.1 Darstellung

Faktoren Objekte ausblenden : Diese Option steuert wie schnell Objekte ausgeblendet werden um die fps zu erhöhen. Es wird empfohlen den Standardwert *1* gesetzt zu lassen. Werte zwischen 0 und 1 führen dazu, dass Objekte später ausgeblendet werden, also länger sichtbar sind. Bei Werten größer als 1 werden Objekte früher ausgeblendet und die fps steigen. Die Option kann für normale Objekte und für Objekte bei welchen *weit sichtbar* definiert ist getrennt gesetzt werden. ab 2.9

Windows 8 Vollbildmodus : Bis Version 2.8.3 war die Performance von Loksim unter Windows 8 im Vollbildmodus auf vielen Systemen im Vergleich zur Performance im Fenstermodus sehr schlecht. Mit Version 2.9 wurde deshalb die Option *Windows 8 Vollbildmodus* eingeführt. Diese Option kann die Vollbildperformance unter Windows 8 erheblich verbessern, bei manchen Systemen führt sie jedoch zu Bildfehlern. Windows 8 Nutzer sollte diese Option bei Bedarf auf dem eigenen System testen. Mit dieser aktivierten Option gibt es jedoch ein anderes Problem: Dialogboxen des Loksim werden im Vollbildmodus mit aktivierter "Windows 8 Vollbildmodus" Option nicht im Vordergrund angezeigt. Dialogboxen wie "Loksim wirklich beenden?" oder "Halt überfahren - Zurück zu Halt?" müssen deshalb "blind" mit den Cursor-Tasten und Enter bedient werden. ab 2.9

### 3.2 Simulation

Doppelklick auf Sifa-Taste startet Wegmessung : Diese Option ist für Loks relevant, welche mit einem Zuglängenzähler / Wegmessung ausgestattet sind. Falls diese Einstellung aktiviert ist, kann die Wegmessung mit zweimaligem Drücken der Sifa-Taste oder zweimaligem Loslassen der realen Sifa-Taste (Joysticksteuerung), gestartet werden. ab 2.8.3

### 3.3 Konfigurationsdateien

#### 3.3.1 Datenpfade

Loksim unterscheidet zwischen zwei grundsätzlichen Orten an denen wichtige Dateien abgelegt werden:

Programmverzeichnis : Hier werden die ausführbaren Programmdateien und einige essentielle sonstige Dateien die unbedingt zum Betrieb benötigt werden abgelegt. Standardmäßig befindet sich dieses Verzeichnis unter Programm-Verzeichnis/Loksim3D

Datenverzeichnis : In diesem Verzeichnis werden Addons/Packages bzw. auch das Loksim-Standardpackage abgelegt. Standardmäßig befindet sich dieses Verzeichnis unter Öffentliche-Dokumente/Loksim3D

Beide Verzeichnisse können bei der erstmaligen Installation von Loksim3D eingestellt werden und mittels "Über Loksim" angezeigt werden. Es ist auch möglich, dass beides auf das gleiche Verzeichnis verweist.

Im selben Verzeichnis wie Loksim3D.exe bzw. LoksimEdit.exe (Programmverzeichnis) wird für diesen Zweck eine Datei paths.ini abgelegt. In dieser wird das Datenverzeichnis angegeben. Hierbei sind sowohl relative als auch absolute Pfadangaben möglich.

Alternativ kann auch "<Registry>" angegeben werden: Dann wird der Pfad aus der Registry ausgelesen, welcher bei der letzten Installation von Loksim3D als Datenverzeichnis angegeben wurde. ab 2.9.3

### 3.4 Besondere Einstellungen

Angepasste Sichtweite Fahrplan : In der Fahrplanauswahl gibt es unter dem Vorschaubild pro Fahrplan eine Option "Angepasst Sichtweite". Wird hier eine Prozentzahl ungleich 100 eingegeben, wird bei diesem Fahrplan die Sichtweite entsprechend vergrößert oder verkleinert. Gibt man beispielsweise "60" an, beträgt beim Fahren dieses Fahrplans sowohl die Sichtweite nah, als auch die Sichtweite fern nur mehr 60% von den in den Optionen eingestellten Sichtweiten. ab 2.9.3

---

## 4 PackageManager

---

### 4.1 Übersicht

Der PackageManager wird zur (De)installation von Loksim Packages verwendet. Loksim Packages besitzen im Normalfall die Dateierweiterung `.l3dpack` (teilweise auch `.zip`).

Im Normalfall wird beim Öffnen einer `.l3dpack` Datei automatisch der PackageManager gestartet. Ein Klick auf Installation startet die Installation. Wird ein Package nicht mehr benötigt, kann es im Tab Packages deinstallieren wieder von der Festplatte gelöscht werden. Dabei werden eventuelle Abhängigkeiten von anderen Packages beachtet und ausschließlich nicht mehr benötigte Dateien gelöscht.

### 4.2 Funktionsweise

- Bei jeder Installation eines Packages protokolliert der Manager, welche Dateien installiert wurden. So kann er bestimmen, welche Dateien von welchen Packages benötigt werden. Bei der Deinstallation eines Packages werden jene Dateien gelöscht, die ansonsten von keinem Package mehr benötigt werden. Dateien, die bereits im Loksim-Verzeichnis existieren, werden niemals gelöscht (es ist im Nachhinein nicht bestimmbar, welche Packages welche Dateien benötigen; bzw. welche Packages überhaupt installiert sind).
- Wie bisher werden bestehende Dateien in den Backup Ordner kopiert, falls sie bei einer Package Installation überschrieben werden. Bei einer Deinstallation werden sämtliche gelöschten Dateien ebenfalls in den Backup Ordner verschoben.
- Bei der Deinstallation von Packages muss man sich im Klaren sein, dass nicht exakt der Zustand "vor der Installation" wiederhergestellt wird: Beispiel: Man installiert das Package A und dann B. Beide installieren die Datei x (im Package B ist die Datei x neuer). Bei der Deinstallation von Package B bleibt jedoch die Datei x, die bei der Installation von Package B kopiert wurde, zurück.
- Über die Optionen ist die Funktion *Installation rückgängig machen* aktivierbar: Im Gegensatz zur Deinstallation eines Package, kopiert diese Funktion die gesicherten überschriebenen Dateien aus dem Backup Verzeichnis zurück in das Loksim-Verzeichnis. Jedoch ist diese Funktion immer nur für das zuletzt installierte Package anwendbar und nicht für früher installierte Packages.

*Wichtig: Deinstallierte Packages können nicht wiederhergestellt werden!*

### 4.3 Deinstallation während Installation

ab 2.8.2

Wie im Abschnitt über Package erzeugen beschrieben wird, können bei der Installation eines neuen Package gleichzeitig ältere Packages deinstalliert werden.

Die Motivation dahinter ist, dass es oftmals neuere Versionen von Packages gibt wo sich die Ordnerstrukturen ändern, manche Objekte nicht mehr gebraucht werden oder Duplikate gelöscht wurden. Installiert der Benutzer diese neue Version, bleiben jedoch die alten, nicht mehr gebrauchten Dateien trotzdem im Loksim Verzeichnis zurück. Bei Führerständen und Fahrplänen kann dies sogar zur Verwirrung des Benutzers führen, bei sämtlichen anderen Dateien bleiben unschöne "Leichen" im Loksim Verzeichnis zurück die nicht mehr gebraucht werden

Wird jedoch beim Erzeugen des Package darauf geachtet, dass sämtliche älteren Versionen des Package bei der Installation der neuen Version gelöscht werden, gibt es solche Probleme nicht. Technisch betrachtet verhält sich die Deinstallation eines Package während der Installation eines anderen Package so, als würde man zuvor manuell die Deinstallation der älteren Packages vornehmen. Jedoch ist der Mechanismus etwas ausgewachsener, sodass wirklich nur jene Dateien gelöscht bzw. kopiert werden, bei denen es tatsächlich auch nötig ist. So sind in der Übersicht der installierten Dateien wirklich nur neuere Versionen zu sehen

*Einschränkungen:* Bei dieser Art von Deinstallation ist es nicht möglich auszuwählen, welche Dateien exakt deinstalliert werden sollen. Packages werden hierbei ganz oder gar nicht deinstalliert. Außerdem bleibt das Prinzip erhalten, dass Deinstallationen nicht rückgängig gemacht werden können! Weder über die Package deinstallieren, noch über die Installation rückgängig machen Funktion

---

## II. Editor

---

---

## 5 Objekte

---

Bis zur Version 2.8.3 war das Objektsystem des Loksims relativ einfach unterteilt: Sämtliche 3D-Objekte waren entweder einfache Objekte (.l3dobj) oder Gruppenobjekte (.l3dgrp), wobei Gruppenobjekte aus beliebig vielen Objekten (.l3dobj) und Schriften bestehen konnten.

Ab Version 2.9 ist das System flexibler gestaltet: Wie bisher sind einfache .l3dobj Objekte und Gruppenobjekte .l3dgrp die "Eckpfeiler" des Objektsystems. Jedoch können Gruppenobjekte nun aus beliebigen Objekte die von Loksime unterstützt werden bestehen. Dies heißt, Gruppenobjekte können wiederum andere Gruppenobjekte beinhalten und auch Objekte in externen Dateiformaten.

### 5.1 Objekte

Objekte (.l3dobj) sind das Grundgerüst für die 3D-Welt. Objekte bestehen aus Punkten und Flächen.

#### 5.1.1 Objekteigenschaften

Textur und Transparenz : nähere Details

Rückseiten sichtbar : Normalerweise ist eine Fläche immer nur von einer Seite sichtbar, wobei die sichtbare Seite anhand der Reihenfolge der Punkte einer Fläche abhängig ist. Wird diese Option gesetzt, sind alle Flächen von beiden Seiten sichtbar. Diese Option sollte nur bei kleinen oder selbstleuchtenden Objekten gesetzt werden, normalerweise sollte explizit eine Rückseite mittels zweiter Fläche eingefügt werden. Nur so ist eine ordentliche Beleuchtung von beiden Seiten möglich

selbst leuchtend : Wird diese Option gewählt wird das Objekt nicht durch das Beleuchtungssystem beleuchtet, dadurch gibt sich ein Effekt als würde das Objekt selbst leuchten. Normalenvektoren sind bei selbst leuchtenden Objekten nicht relevant

Objekt dreht mit dem Betrachter mit : Wird diese Option gesetzt, dreht sich das Objekt immer so, dass die Vorderseite zum Betrachter zeigt. So können flache Objekte aus gewisser Entfernung wie plastische Objekte wirken.

Normalenvektor pro Fläche : Ist diese Option gewählt, werden die Normalenvektoren nicht wie sonst pro Punkt eingegeben, sondern pro Fläche (nähere Details) ab 2.9



### 5.1.2 Punkte

Jeder Punkt hat eine bestimmte Position im Raum und kann einen Normalenvektor enthalten.

### 5.1.3 Flächen

Flächen bestehen aus drei oder mehr Punkten in *einer Ebene*. Bei der Definition einer Fläche dürfen keine Löcher entstehen. Mit einem Rechtsklick auf eine Fläche kann automatisch die Rückseite zu der gewählten Fläche eingefügt werden. ab 2.9

Fehler (nicht-planar, zu wenige Punkte, ungültige oder doppelte Koordinate) bei der Fläche werden beim Editieren der Fläche angezeigt. ab 2.9.3

Diese Meldungen sollten beim Bau neuer Objekte und bei der Überarbeitung bestehender Objekte unbedingt beachtet werden. Fehlerhaft definierte Flächen werden "so gut wie möglich" dargestellt. Das Ergebnis entspricht dabei nicht immer den Erwartungen und kann von Version zu Version unterschiedlich sein. Diese Fehler werden auch in die Logdateien eingetragen. Somit ist es beispielsweise möglich, eine gesamte Strecke nach solch fehlerhaften Objekten abzusuchen.

### 5.1.4 Normalenvektoren

ab 2.9

Normalenvektoren werden zur Beleuchtung von Objekten verwendet. Bei "eckigen Oberflächen" sollten die Normalenvektoren immer *senkrecht* auf die Fläche stehen. Bei "runden Oberflächen" kann durch die richtige Wahl der Normalenvektoren der Eindruck der runden Oberfläche verbessert werden. Diese Anleitung kann keine genaue Einführung in Normalenvektoren geben, erfahrene Objektbauer können aber an vielen Stellen im Internet genauere Details dazu finden. Normalenvektoren sind kein Konzept des Loksims, sondern werden generell in der 3D-Modellierung und -Darstellung genutzt.

Da Loksims bisher recht gut mit komplett falsch gesetzten Normalenvektoren umgegangen ist, hat sich beim Loksims-Objektbau eine gewisse Schlampigkeit bzgl. Normalenvektoren entwickelt. Ab Version 2.9 gibt es deshalb eine gravierende Änderung: Standardmäßig wird bei Objekten die Option *Normalenvektor pro Fläche* gesetzt. Wird diese Option verwendet, kann es keine unterschiedlichen Normalenvektoren pro Fläche geben. Außerdem steht über das Menü - Bearbeiten die Funktion *Normalenvektoren automatisch berechnen (Standard)* zur Verfügung. Diese erzeugt Normalenvektoren, die senkrecht auf die jeweilige Fläche stehen. Für die meisten Objekte sollte diese Option ausreichen.

Für runde Objekte muss diese Einstellung jedoch deaktiviert werden und die Normalenvektoren wie bisher pro Punkt definiert werden. Die Normalenvektoren für runde Objekte können mittels der Funktion Bearbeiten - *Normalenvektoren automatisch berechnen (rundes Objekt)* automatisch berechnet werden. ab 2.9.3

Da Loksिम in Zukunft ein verbessertes Beleuchtungssystem bekommen wird, gibt es folgende Regelung: Bei Objekten die mit Version 2.9 oder neuer erstellt oder abgeändert wurden wird angenommen, dass sie mit korrekten Normalenvektoren ausgestattet sind. Ältere Objekte werden vorraussichtlich anders verarbeitet werden, sodass falsche Normalenvektoren nicht so sehr ins Gewicht fallen.

### 5.1.5 Zoomen/Verschieben

ab 2.9

Über das Bearbeiten Menü stehen die Funktionen *Punkte verschieben/zoomen* und *Objekt am Nullpunkt zentrieren* zur Verfügung. Bei erster kann gewählt werden welche Punkte verschoben/-gezoomt werden sollen und um welchen Wert die Änderung erfolgen soll. Bei zweiter Funktion wird automatisch der Nullpunkt des Objekts berechnet und der Dialog zum Verschieben aller Punkte geöffnet. Hier sind bereits die Werte eingetragen welche nötig sind, um das Objekt in den Nullpunkt zu verschieben. Die Werte können vor dem Verschieben noch angepasst werden.

## 5.2 Gruppenobjekte

ab 2.9

Gruppenobjekte können beliebige andere von Loksिम unterstützte Objekte enthalten. Für jedes Objekt kann die Position, Rotation und Skalierung für jede Achse eingestellt werden. Außerdem können einzelne Objekte mit der Sichtbarkeitssteuerung ein- und ausgeblendet werden.

Die 2D-Vorschau unterstützt derzeit nur loksिम-eigene Formate, externe Objektformate werden in der 2D-Vorschau nicht angezeigt.

### 5.2.1 Verschieben

ab 2.9

Über das Bearbeiten Menü stehen die Funktionen *Punkte verschieben* und *Objekt am Nullpunkt zentrieren* zur Verfügung. Bei erster kann gewählt werden welche Objekte verschoben werden sollen und um welchen Wert die Änderung erfolgen soll. Bei zweiter Funktion wird automatisch der Nullpunkt des Objekts berechnet und der Dialog zum Verschieben aller Objekte geöffnet. Hier sind bereits die Werte eingetragen welche nötig sind um das Gesamtobjekt in den Nullpunkt zu verschieben. Die Werte können vor dem

Verschieben noch angepasst werden.

## 5.3 Externe 3D-Objektmodellformate

ab 2.9

Es besteht die Möglichkeit alle von Assimp<sup>1</sup> unterstützten externe Objektformate zu verwenden. Allerdings ist das Objektsystem immer noch auf .l3dgrp und .l3dobj Dateien ausgerichtet. Alles was also nicht mittels .l3dgrp oder .l3dobj Dateien möglich ist, kann derzeit auch nicht durch externe Formate umgesetzt werden.

Je nach Bedarf der Objektbauer wird die Unterstützung externer Dateiformate noch erweitert werden.

### 5.3.1 Konvertieren externer 3D-Objektmodellformate

Über die Kommandozeile lassen sich alle unterstützen Dateiformate in Loxsim Gruppen(objekte) umwandeln. Dafür muss der LoxsimEdit in der Konsole folgenderweise aufgerufen werden:

```
LoxsimEdit -convert <Pfad Quellobjekt> <Pfad Zielobjekt>
```

Der "Pfad Zielobjekt" sollte dabei auf eine nicht existierende Datei in einem *leeren* Ordner verweisen. Beim Konvertieren werden möglicherweise mehrere Dateien erstellt, bestehende Dateien die zufällig den gleichen Namen tragen werden dabei ohne Rückfrage überschrieben.

## 5.4 Gleise

Gleise sind spezielle - von Loxsim selbst erzeugte - Objekte. Das Aussehen der Gleise wird mittels .l3drail Dateien beschrieben.

Transparenz : Wie bei herkömmlichen Objekten können auch Gleise teilweise transparent sein. Nähere Details dazu findet man im Abschnitt über Transparenz bei Objekten ab 2.9.3

Normalenvektoren senkrecht : Bis Version 2.9.2 wurden sämtliche Normalenvektor bei den generierten Gleise auf 0/1/0 gestellt. Wird die Option "Normalenvektoren senkrecht" aktiviert, werden die Normalenvektoren korrekt berechnet und stehen immer senkrecht zur jeweiligen Fläche. ab 2.9.3

Keine 3D-Darstellung : Ein beliebter Trick für die Landschaftdarstellung und Objektpositionierung ist das Anlegen unsichtbarer Gleise. Wird diese Option gewählt ist dies ein Hinweis für das Programm, dass entsprechende Gleise nicht für die Darstellung in 3D verwendet werden, sondern nur Hilfgleise für Objekte bzw.

---

<sup>1</sup> [http://assimp.sourceforge.net/main\\_features\\_formats.html](http://assimp.sourceforge.net/main_features_formats.html)

Landschaft sind. Dies führt zur Reduzierung von Ladezeiten  
und Verbesserung der fps ab 2.9.3

## 5.5 Textur

Texturen sind Bilder welche auf auf die dreidimensionalen Objekte aufgebracht werden können, um die Darstellungsqualität erheblich zu verbessern. Loxsim unterstützt die Bildformate *BMP*, *PNG* und *TGA* für Texturen.

### 5.5.1 Transparenz

Zusätzlich können Objekte mit einer Transparenzkomponente ausgestattet werden, um (halb)durchsichtige Oberflächen umzusetzen. Dafür werden folgende Techniken angeboten:

nicht transparent : Unabhängig von der verwendeten Textur wird das Objekt vollständig undurchsichtig dargestellt

Schwarz ist transparent : Alle Bereich die komplett schwarz sind (0x000000), werden transparent dargestellt (für neue Objekte nicht mehr empfohlen).

Transparenzfarbe ist die Farbe des Pixels links/oben : Alle Bereiche welche die exakt gleiche Farbe wie das Pixel in der linken, oberen Ecke haben, werden transparent dargestellt (für neue Objekte nicht mehr empfohlen)

Transparenz aus Bitmap (Weiß undurchsichtig...) : Für diesen Typ muss eine zweite Bilddatei angegeben werden. Die Transparenz wird dann mittels dieser zweiten Datei bestimmt, wobei Weiß undurchsichtig ist, schwarz komplett transparent und die Graustufen dazwischen halbdurchsichtige Bereiche ergeben. (für neue Objekte nicht mehr empfohlen)

Transparenz aus Alphakanal - alle Transparenzwerte möglich : Die Transparenz wird aus dem Alphakanal der Textur ausgelesen. Dies ist nur bei Bildformaten möglich, welche einen Alphakanal unterstützen. ab 2.8.3

Transparenz aus Alphakanal - nur sichtbar/unsichtbar : Die Transparenz wird wie bei der vorigen Option aus dem Alphakanal ausgelesen. Es werden jedoch keine beliebigen Transparenzwerte unterstützt, sondern nur eine Unterscheidung zwischen sichtbaren und unsichtbaren Pixel vorgenommen. Wenn möglich sollte diese Option aus Performancegründen der anderen Alphakanal-Option vorgezogen werden. ab 2.8.3

## 5.6 Texturnutzung optimieren

ab 2.9.2

Diese Funktion ermöglicht es die Texturen von beliebig vielen (Gruppen)objekten in möglichst wenig Texturen zusammenzufassen. In vielen Fällen wird das einfach nur ein einzelnes Gruppenobjekt sein, bei dem man die Texturen der Bestandteile zusammenfassen möchte, man kann aber auch mehrere Objekte auswählen. Die Texturen die dabei entstehen sind vom Platzbedarf nicht immer ganz perfekt, aber doch ganz ordentlich. Die Ergebnistextur ist immer eine PNG-Datei, wo bei Bedarf der Alphakanal entsprechend gesetzt ist.

Objekt hinzufügen : Über diese Schaltfläche und über das Kontextmenü in der Objektliste können (Gruppen)Objekte hinzugefügt werden, deren Texturen in möglichst wenigen Texturen zusammengefasst werden sollen. Über das Kontextmenü in der Liste können hinzugefügte Objekte wieder gelöscht werden.

Zielordner : Im Zielordner werden die neuen Teilobjekte sowie die generierten Texturen gespeichert. Dieser Ordner sollte leer sein, da etwaige bestehende Dateien ohne Rückfrage überschrieben werden.

Maximale Texturegröße : Die maximale Texturgröße der neu generierten Texturen. In den meisten Fällen sollte man hier die Standardauswahl von 1024 benutzen.

Basis Texturname : Diese optionale Einstellung wird für die Namensgebung der neu generierten Texturen verwendet.

Ein klassisches Einsatzgebiet dieser Funktion ist, wenn man sich beim Bauen eines ganz neuen Objekts keine Gedanken über die optimale Texturnutzung machen möchte und einfach mit lauter Einzeltexturen los baut. Am Ende muss dann nur diese Funktion angewendet werden und man hat eine gute Zusammenfassung der Einzeltexturen in weniger Texturen.

### 5.6.1 Empfehlungen

- Der Platz auf einer Textur sollte nach Möglichkeit optimal ausgenutzt werden. Bei einem Objektset oder Gruppenobjekten können sich dafür auch mehrere Objekte eine Textur teilen.
- Texturen sollten eine Höhe und Breite haben, welche einer Zweierpotenz ( $2^n$ ) entspricht. Typische Werte sind 2, 4, 8, 16, 32, 64, 128, 256, ...
- Die Texturgröße sollte sich an der maximalen Größe, mit der ein Objekt in der Szene dargestellt werden soll, orientieren. Für ein einfaches Signallicht, reicht deshalb möglicherweise

schon eine 8x8px große Textur. Für sehr große Objekte wie beispielsweise Kulissen können die Texturen auch entsprechend groß werden (1024px oder im Extremfall 2048px), um eine hohe Detailgenauigkeit zu erreichen.

- Anstatt extrem große Texturen zu verwenden, ist es oftmals besser, ein Objekt in Teilobjekte aufzuteilen. Objekttexturen welche größer als 1024px sind, sollten nur extrem selten genutzt werden.
- Für die meisten Objekte sollten Texturen mit 256x256px oder 512x512px ausreichen.
- Für neue Objekte sollten "Transparenz aus Bitmap", "Transparenzfarbe ist die Farbe des Pixels links/oben" und "Schwarz ist transparent" *nicht mehr* verwendet werden. ab 2.8.3
- Die in älteren Versionen notwendigen Größenanpassungen bei transparenten Texturen sind *nicht mehr* notwendig. ab 2.8.3

### 5.6.2 Einfluss der Texturen auf Performance

#### Texturgröße

Ein wichtiger Faktor für die Performance (und damit die erreichbaren fps) ist die Anzahl der so genannten "Draw Calls." Sehr einfach gesagt ist ein Draw Call der Befehl an die Graphikkarte eine (fast) beliebige Anzahl an Dreiecken mit einer bestimmten Textur an einem bestimmten Ort zu zeichnen. Insbesondere bei der derzeit von Loxsim verwendeten Graphikschnittstelle spielen diese Draw Calls eine wichtige Rolle. Dies bedeutet im Umkehrschluss, dass für die Anzeige von zwei Objekten die eine unterschiedliche Textur nutzen auch zwei Draw Calls notwendig sind.

Unter Berücksichtigung dieser Tatsachen ist es sinnvoll mehrere kleinere Texturen auf einer größeren Textur zusammenzufassen. Insbesondere bei Objekten die (fast) immer gemeinsam angezeigt werden: Beispielsweise die Teilobjekte eines Gruppenobjekts. Jedoch kann diese Optimierung auch für Texturen von mehreren Gruppenobjekten die fast immer "nebeneinander" angezeigt werden, Sinn machen. Hierzu gehören zB mehrere Gebäude eines charakteristischen Bahnhofs. Da durch das Zusammenfassen von Texturen eine größere Textur entsteht, sollte man jedoch Texturen von Objekten die häufig nicht miteinander angezeigt/aufgestellt werden, nicht zusammenfassen. Denn kleine Texturen sind sowohl bei der Darstellung als auch beim Laden effizienter.

Andererseits sollten Texturen stets so klein wie möglich sein und den vorhandenen Platz möglichst gut ausnutzen. Auch wenn der Festplattenspeicher (und meist auch der Arbeitsspeicher) heute

meistens groß genug ist (Loksim kann als 32Bit Anwendung nicht mehr als 4GB Hauptspeicher nutzen), so sind die Speicher auf - insbesondere integrierten - Graphikkarten nicht immer so groß. Zusätzlich ist zu beachten, dass kleinere Texturen schneller von einem Speicher in den anderen übertragen werden können. Die Wahl einer sinnvollen Texturgröße ist also immer noch wichtig.

Zum einen ist dabei die Überlegung wichtig, wie groß das Objekt für welches die Textur gedacht ist, in der Simulator dann tatsächlich angezeigt wird. Eine Textur für ein Signallicht muss deshalb nur wenige Pixel breit sein. Andere Texturen wie zB die Texturen die für die Himmelsdarstellung verwendet werden, sollten auch für FullHD (1920x1080px) genug Details beinhalten. 4K Monitore können derzeit von Loksim nicht voll ausgenutzt werden und müssen deshalb beim Objektbau (noch) nicht beachtet werden. Eine Hausmauer die beispielsweise hin und wieder den halben Bildschirm des Simulators ausfüllt, kann also ruhig eine Breite von 1024px aufweisen. Größere Texturen sollte man eher vermeiden und bei Bedarf eher das Objekt zerteilen: Eine Mauer wird zB selten den ganzen Bildschirm ausfüllen und sicherlich oftmals auch nur zum Teil dargestellt werden. Dann ist es effizienter, zwei halb so große Texturen zu benutzen und nicht eine einzige die doppelt so groß ist. Auch ist die Standardeinstellung der maximalen Texturgröße im Loksim derzeit 1024px. Ist eine Textur größer als diese Maximalgröße, wird die Textur beim Laden auf die Maximalgröße verkleinert.

Zusätzlich ist zu beachten, dass zur Laufzeit intern immer mit Texturen die Zweierpotenzen als Seitengröße haben gearbeitet wird. Eine Textur der Größe 200x70px wird also beispielsweise beim Laden auf eine Textur mit 256x128px vergrößert. Die Textur wird dabei nicht tatsächlich "vergrößert", sondern es wird ein - nicht benutzter - Rand eingefügt. Das bedeutet, dass eine Textur der Größe 200x70px und eine mit 256x128px zur Laufzeit gleich viel Speicherplatz benötigt. Man könnte also entweder die Textur der Größe 200x70px auf 256x128px vergrößern, um mehr Details darstellen zu können oder man verwendet den "nicht genutzten Rand" für ein zweites/weiteres Objekt. Es sei dabei gesagt, dass Texturen die keine Zweierpotenzen als Seitenlängen benutzen "nur" den Nachteil der Platzverschwendung haben.

## Transparenz

Für transparente Objekte sollten die Optionen "schwarz ist transparent", "Transparenzfarbe ist die Farbe des Pixels links/unten" und "Transparenzfarbe aus Bitmap" nicht mehr verwendet werden. Diese sind beim Laden aufwändiger zu behandeln wie die zwei neuen Methoden. Transparenz sollte bei neuen Objekten also nur mehr durch den Alphakanal in der Textur umgesetzt werden. Dafür stehen zwei Einstellungen zur Verfügung:

Transparenz aus Alphakanal – nur sichtbar/unsichtbar : Diese Option sollte gesetzt werden, falls die Textur nur vollkommen durchsichtige bzw. undurchsichtige Bereiche enthält.

Transparenz aus Alphakanal – alle Transparenzwerte möglich : Diese Option muss gesetzt werden, falls die Textur auch halbtransparente Bereiche enthält.

Ist man sich nicht sicher, sollte "Transparenz aus Alphakanal – alle Transparenzwerte möglich" gesetzt werden. In einer zukünftigen Loxsim-Version könnten Objekte mit "Transparenz aus Alphakanal – nur sichtbar/unsichtbar" jedoch eventuell schneller dargestellt werden.

*Achtung:* Derzeit gibt es keine Unterscheidung zwischen diesen beiden Optionen. Dies muss jedoch in Zukunft nicht so sein, und wenn bei einem Objekt fälschlicherweise "Transparenz aus Alphakanal – nur sichtbar/unsichtbar" eingestellt ist, kann es zu Fehldarstellungen kommen.

Bei Objekten mit transparenten und nicht-transparenten Flächen (zB ein Fahrzeug mit nicht-transparenter Karosserie und transparenten Scheiben) spielt die Reihenfolge der Flächen eine Rolle. (Halb-) Transparente Flächen sollten dabei unter den undurchsichtigen Flächen angelegt werden bzw. dorthin verschoben werden.

## 5.7 Logische Ausdrücke

Bei Streckenobjekten, Gruppenobjekten und Streckensounds können logische Ausdrücke definiert werden, welche bestimmen ob das Objekt angezeigt wird oder nicht (bzw der Sound abgespielt wird oder nicht).

In logischen Ausdrücken können Variablen (Operanden) mittels Operatoren verknüpft werden: Wird dieser Ausdruck zu einem Wert gleich 0 ausgewertet, so wird der gesamte Ausdruck als falsch gewertet. Ist der berechnete Wert ungleich 0, so ist der gesamte Ausdruck wahr.

Dieses Verhalten, die möglichen Operatoren und Priorität entsprechen der C/C++ Programmiersprache.

### 5.7.1 Operanden (Variablen)

Konstante

Es können folgende Arten von Konstanten als Operatoren verwendet werden:

Ganzzahlen : Ganze Zahlen im Bereich  $-(2^{31}) + 2$  bis  $2^{31} - 1$

Zeitpunkte : Zeitangaben im Format hh:mm:ss oder hh:mm (Stunden:Minuten:Sekunden)  
Diese Art von Angabe wird bei der Auswertung in Sekunden nach 00:00 Uhr umgerechnet.



Zeichenketten : Zeichenketten - auch Strings oder alphanumerische Werte genannt - sind mit einem Anführungszeichen am Beginn und Ende gekennzeichnet («Wert>»). Auch Zeichenketten werden mit einer Hash-Funktion in ganze Zahlen umgewandelt. Es kann deshalb in sehr unwahrscheinlichen Fällen vorkommen, dass zwei unterschiedliche Zeichenketten in die gleiche Ganzzahl umgewandelt werden. Bei Zeichenketten macht nur der Vergleich auf (Un)gleichheit Sinn; Vergleiche wie welche Zeichenkette ist die "größere" von zwei werden von Loksim nicht unterstützt und das Ergebnis kann sich von Version zu Version ändern. ab 2.9.2

## Variablen

Zusätzlich kann auf diverse Daten per Namen zugegriffen werden. Diese Daten werden immer mittels *Namensraum::Variablenname* angesprochen.

Derzeit sind folgende Namensräume bzw Variablen definiert:

**Sim** : Dieser Namensraum enthält alle Eigenschaften die in früheren Versionen über die *nur (un)sichtbar bei* Optionen verfügbar waren. ab 2.9

Die folgende Auflistung enthält derzeit nur einen Bruchteil der verfügbaren Variablen, alle möglichen Variablen sind über das entsprechende Auswahlfeld im Objekteditor ersichtlich

**VSigKennzahlKleiner** : Diese Variable ist TRUE, wenn das Haupt- und das Vorsignal am gleichen Standort Kennziffern haben und die Vorsignalkennziffer kleiner als die (Haupt-)Signalkennziffer ist. In allen andern Fällen ist VSigKennzahlKleiner FALSE.

**WetterVars** : Ermöglicht den Zugriff auf vom Benutzer definierte Variablen in Wetterdateien

**FahrplanVars** : Ermöglicht den Zugriff auf vom Benutzer definierte Variablen in Fahrplänen

**FahrplanDaten** : Ermöglicht den Zugriff auf diverse Daten des Fahrplans

**Abfahrtszeit("<halt>")** : geplante Abfahrtszeit des Halts <halt> in Sekunden

**Ankunftszeit("<halt>")** : geplante Ankunftszeit des Halts <halt> in Sekunden

**Halt("<halt>")** : Wahr falls der Zug an dieser Haltestelle hält ab 2.9

**BedarfshaltBahnsteig("<halt>")** : Wahr falls der Halt ein Bedarfshalt ist und der Haltanzeiger am Bahnsteig aufleuchtet ab 2.9

**BedarfshaltZug("<halt>") :** Wahr falls der Halt ein Bedarfshalt ist und der Haltanzeiger im Führerstand aufleuchtet ab 2.9

**LastHalt :** Letzter Halt im Fahrplan der keine Zugfolgestelle bzw. Betriebshalt ist ab 2.9.3

Innerhalb der Anführungszeichen muss der exakte Name des Halts aufgeführt werden. Sollte der Haltname selbst ein Anführungszeichen enthalten, muss dieses durch \ ersetzt werden. Der Halt St. Pölten "Hbf" wird damit zu St. Pölten \ "Hbf\"

**WetterDaten :** Ermöglicht den Zugriff auf vom Programm berechnete Daten zum Wetter

**HelligkeitProzent :** Helligkeit im Bereich 0-100

**Simulation :** Diverse Eigenschaften der Simulation

**ZeitInSekunden :** Seit 00:00 vergangene Sekunden

**Sonstige :** Sammelkategorie für den Rest

**Zufall :** Liefert einen zufälligen Wert im Bereich [0;Sehr große Ganzzahl[

**Zuglaenge :** Liefert die Länge des Zugs in Metern (inklusive Triebfahrzeug)

**ZufallGruppenObjekt :** Liefert einen zufälligen Wert im Bereich [0;Sehr große Ganzzahl[, welcher für das gesamte Gruppenobjekt gleich ist. Mit dieser Variable ist es möglich, mehrere Teilobjekte eines Gruppenobjekts zufällig ein- oder auszublenden.

Der Einsatz dieser Variablen macht aus diesem Grund auch nur innerhalb eines Gruppenobjekts Sinn

Wird auf eine Variable zugegriffen, die nicht definiert ist, so wird 0 (falsch) zurückgeliefert

**Testen im LoksimEdit** Zu Testzwecken können im LoksimEdit unter Ansicht - Erweiterte 3D-Ansichtsoptionen oder Strg+Q *sämtliche* (auch von der Simulation vorgegeben, wie ZeitInSekunden) Variablen mit benutzerdefinierten Werten belegt werden. Damit kann man Testen, wie die Anzeige mit dieser Variablenbelegung aussieht

## 5.7.2 Operatoren

Operatoren verknüpfen die Operanden miteinander um das Endergebnis zu berechnen. Auflistung der verfügbaren Operatoren nach

*absteigender* Priorität angeordnet (gleiche Hintergrundfarbe bedeutet gleiche Priorität):

Opera- tor	Bedeutung	Alternative Syntax
!	logisches <i>Nicht</i> , Negation: Wandelt einen Wert ungleich 0 in 0 um bzw umgekehrt	not
*	<i>Multiplikation</i>	
/	ganzzahlige <i>Division</i>	
%	<i>Modulo</i> (Rests der Division)	
+	<i>Addition</i>	
-	<i>Subtraktion</i>	
<	logischer Vergleich: <i>kleiner</i>	
<=	logischer Vergleich: <i>kleiner oder gleich</i>	
>	logischer Vergleich: <i>größer</i>	
>=	logischer Vergleich: <i>größer oder gleich</i>	
==	logischer Vergleich auf <i>Gleichheit</i>	
!=	logischer Vergleich <i>ungleich</i>	<>
&&	logische Verknüpfung <i>und</i> ; liefert wahr, falls beide Operanden wahr sind	and
	logische Verknüpfung <i>oder</i> ; liefert wahr, falls einer der Operanden wahr ist	or

Tabelle 5.1: Operatoren in logischen Ausdrücken

### 5.7.3 Funktionen

Funktionen::TimeDif()

Diese Funktion berechnet die Zeitdifferenz zwischen zwei Zeitpunkten und beachtet dabei die Möglichkeit des Tageswechsels. Die genau Logik der Funktion lautet:

```
int TimeDif(Zeit1, Zeit2)
{
    Falls (Zeit1 < Zeit2)
    {
        Ergebnis = Zeit1 + SEKUNDEN_PRO_TAG - Zeit2
    }
    Sonst
    {
        Ergebnis = Zeit1 - Zeit2
    }
    return Ergebnis
}
```

### 5.7.4 Beispiele

Um das ganze etwas klarer zu machen, ein paar Anwendungsfälle:

Ein Objekt soll nur zwischen 22:00 und 05:00 Uhr angezeigt werden, oder wenn die Helligkeit kleiner als 20 % ist:

```
Simulation::ZeitInSekunden >= 22:00 ||  
    Simulation::ZeitInSekunden <= 05:00 ||  
    WetterDaten::HelligkeitProzent < 20
```

Ein Sound soll nur zur vollen Stunde abgepielt werden, aber nicht nach 20:00 bzw vor 06:00 Uhr:

```
Simulation::ZeitInSekunden % (60 * 60) <= 2 &&  
    !(Simulation::ZeitInSekunden > 20:00 ||  
    Simulation::ZeitInSekunden < 06:00)
```

Eine optimierte, gleichwertige Version wäre:

```
Simulation::ZeitInSekunden % 3600 <= 2 &&  
    Simulation::ZeitInSekunden <= 20:00 &&  
    Simulation::ZeitInSekunden <= 06:00 )
```

Erklärung: Zu jeder vollen Stunde ist die ZeitInSekunden durch 3600 ohne Rest teilbar. Dies bedeutet ZeitInSekunden % 3600 ergibt 0. Wenn man nun dies exakt mit 0 vergleicht, würde der Sound nur abgespielt, falls es exakt die volle Stunde ist. Für uns ist jedoch ein kleiner Spielraum von 2 Sekunden ok, also wird mit <= 2 geprüft, sodass der Sound auch 1 oder 2 Sekunden nach der vollen Stunde gestartet werden kann

Ein Objekt soll mit 60%iger Wahrscheinlichkeit angezeigt werden: Sonstige::Zufall % 100 < 60 Erklärung: Zufall liefert eine zufällige Zahl im Bereich 0 bis sehr große Zahl. Rechnet man diese Zahl modulo 100 (Rest der Division der Zahl durch 100) bekommt man eine zufällige Zahl im Bereich [0;100[ In 60% der Fälle ist diese Zahl kleiner als 60, und in den anderen 40 Prozent größer gleich 60. Also hat man genau die 60%ige Wahrscheinlichkeit

Differenz zur Abfahrtszeit des Halts *h0 "ui"*

```
Funktionen::TimeDif(  
    FahrplanDaten::Abfahrtszeit("h0 \"ui\""),  
    Simulation::ZeitInSekunden) / 60)
```

Ein Objekt soll nur angezeigt werden, falls der - händisch in den Fahrplanvariablen - gesetzte Endbahnhof "Hintertupfingen" ist: ab 2.9.2

```
FahrplanVars::Endbahnhof == "Hintertupfingen"
```

### 5.7.5 Einfluss auf Performance

Bei großflächigem Einsatz von logischen Ausdrücken, sollte man etwaige Auswirkungen auf die Performance bedenken:

- Objekte bei denen *Objekt in Sichtweite ändert Sichtbarkeit nicht* gesetzt ist, haben kaum Einfluss auf die Performance: Solche Ausdrücken müssen während der Simulation nur ein einziges Mal berechnet werden und sind anschließend fixiert
- Auch konstante Ausdrücke (kein Zugriff auf Variablen außer *Sonstige::Zufall*) müssen nur ein einziges Mal berechnet werden
- Sonstige logische Ausdrücke bei Objekten werden regelmäßig (mehrmals pro Sekunde) neu berechnet, solange sich die Objekte in sichtbarer Entfernung befinden
- Nicht-konstante, logische Ausdrücke bei Sounds, werden regelmäßig (mehrmals pro Sekunde) neu berechnet, solange sich die Lok im Gültigkeitsbereich des Sounds befindet
- Es werden keinerlei Optimierungen der Ausdrücke vorgenommen!

## 5.8 Sichtbarkeitssteuerung

ab 2.9

Mit Hilfe der Sichtbarkeitssteuerung ist es möglich, (Teil-)Objekte anhand bestimmter "Umgebungseigenschaften" ein- und auszublenden. Die Sichtbarkeitssteuerung wurde mit Version 2.9 grundlegend überarbeitet und vereinheitlicht. Vor Version 2.9 konnte die Sichtbarkeit eines Objekts auf drei unterschiedliche Arten gesteuert werden: Mittels "Nur sichtbar" bei bzw. "nur unsichtbar bei" Ausdrücken und mit Hilfe der "Dynamischen Sichtbarkeitssteuerung." Diese drei Varianten wurden mit Version 2.9 zur allgemein verwendbaren *Sichtbarkeitssteuerung* vereinheitlicht. Alte Objekte werden dabei automatisch beim Laden in das neue Format konvertiert und müssen nicht extra bearbeitet werden.

### 5.8.1 Sichtbarkeitssteuerung in Gruppenobjekten

Teile eines Gruppenobjekts (Objekte und Fonts) können mit der Sichtbarkeitssteuerung ein- bzw. ausgeblendet werden. Dafür wird in das Feld *Sichtbarkeitssteuerung* ein Logischer Ausdruck eingetragen. Dabei können alle Operanden die im Abschnitt der logischen Ausdrücke beschrieben sind, eingesetzt werden. Zusätzlich können beliebigen Variablen die mit *Str::* beginnen eingesetzt werden. Für diese kann vom Streckenautor in der Strecke ein Wert gesetzt werden.

Wird die Option *Objekt in Sichtweite ändert Sichtbarkeit nicht* aktiviert, wird die Sichtbarkeit des Objekts in der Nähe der aktuellen Kameraposition nicht geändert. In vielen Fällen ist diese Option sinnvoll, um zu verhindern, dass ein Objekt plötzlich verschwindet oder aus dem Nichts auftaucht. Allerdings ist es für bewusst

dynamische Objekte wie Signallichter oder Bahnschranken nicht sinnvoll, diese Eigenschaft zu aktivieren.

Für das Testen der Sichtbarkeitsausdrücke steht ein Eigenschaftseditor zur Verfügung, welcher in der 3D-Vorschau mittels dem mit *E* gekennzeichneten Button geöffnet werden kann.

### 5.8.2 Sichtbarkeitssteuerung Streckenobjekte

Im Streckeneditor kann ganzen Streckenobjekten ein Sichtbarkeitsausdruck zugewiesen werden. Hierbei sind ebenfalls die gleichen Operatoren und Variablen wie bei den logischen Ausdrücken beschrieben einsetzbar. Einige wenige Ausnahmen sind in der Auflistung explizit gekennzeichnet. Im Gegensatz zu der Sichtbarkeitssteuerung in Objekten macht es hier keinen Sinn, Variablen die mit *Str::* beginnen einzusetzen.

Es ist zu beachten, dass bei Streckenobjekten gesetzte Sichtbarkeitsausdrücke für jedes enthaltene Objekt einzeln gelten. Beispielsweise werden mit dem Sichtbarkeitsausdruck *Sonstige::Zufall % 100 < 40* zufällig enthaltene Objekte aus oder eingeblendet. Es werden *nicht* gleichzeitig alle Objekte ein- oder ausgeblendet. Um diesen Effekt zu erreichen, müssen die Objekte in ein Gruppenobjekt gepackt werden, welches dann gesamt ein- oder ausgeblendet werden kann.

### 5.8.3 Sichtbarkeitssteuerung über Eigenschaften der Objekte

Bei Streckenobjekten kann für jedes Objekt welches *Str::* Variablen im Sichtbarkeitsausdruck enthält ein Eigenschaftsfenster geöffnet werden. In diesem gibt es für jede Eigenschaft drei Möglichkeiten:

Sichtbarkeitsausdruck : Es wird ein ganzer Sichtbarkeitsausdruck gesetzt, welcher die gleichen Variablen wie Sichtbarkeitsausdrücke bei Streckenobjekten enthalten darf.

Nein/Ja : : Die Eigenschaft wird mit ja/nein (wahr/falsch) fix gesetzt

Aus Variable : : Die Eigenschaft wird aus diversen zur Verfügung stehenden Werten aus der Simulation gefüllt:

Streckenhektometer : Formatierter String für Anzeige des Streckenhektometers auf Hektometertafeln

Streckenmeter : Position des Objekts in Metern

Weichenstellung : Anzeige der Weichenstellung

Zufallsgenerator : Es wird mittels Zufall bestimmt ob die Eigenschaft mit 0 oder 1 (falsch oder wahr) gesetzt werden soll. Eine genauere Steuerung des Zufalls ist mittels Sichtbarkeitsausdruck und *Sonstige::Zufall* möglich

Bahnuebergang : Eigenschaft für Bahnübergänge

Nachdem in Dynamischen Schriften über die *Expr*-Funktion ganze logische Ausdrücke verwendet werden können, können dort verwendete *Str::* wie für die Sichtbarkeitssteuerung in im Eigenschaftsfenster gesetzt werden.

## 5.9 Dynamische Schriften

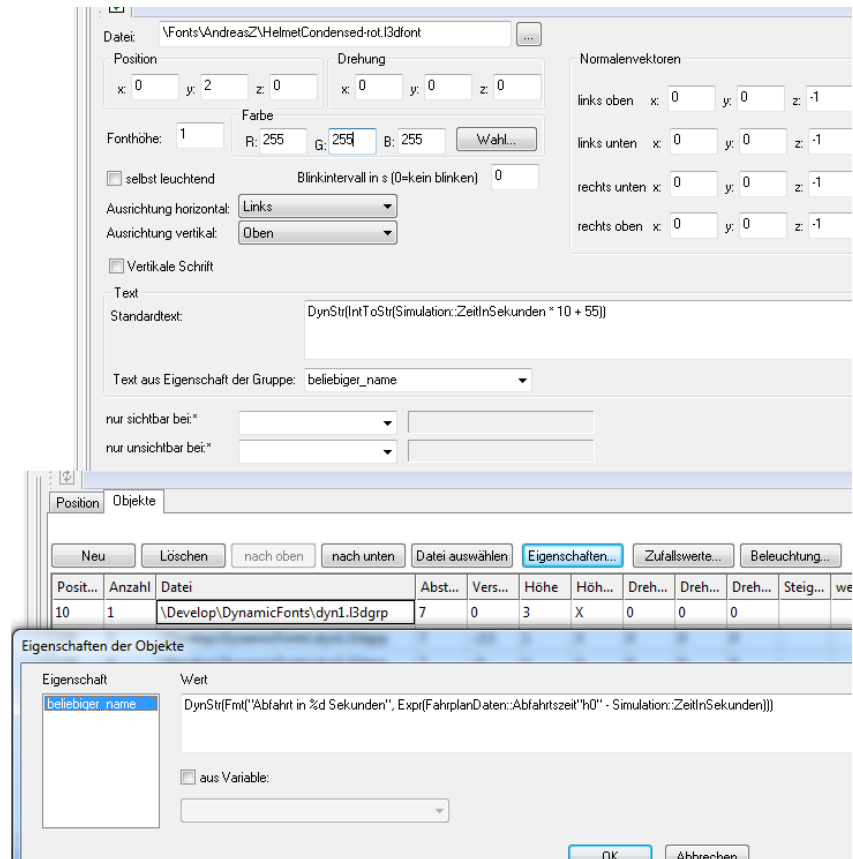


Abbildung 5.1: Möglichkeiten dynamische Schriften zu setzen (oben Gruppenobjekt, unten Streckeneditor)

Mit dynamischen Schriftarten ist es möglich, den Inhalt von Fonts dynamisch oder anhand statischer Variablen zu steuern.

Die Ausdrücke können dabei als Standardtext bei Fonts in Gruppenobjekten eingestellt werden. Oder man setzt den entsprechenden Ausdruck im Streckeneditor über die Eigenschaften des Objekts.

Dynamische Schriften haben immer die Form `DynStr()`. Abbildung 5.1 zeigt zwei Beispiele für die Möglichkeiten Dynamischen Schriften zu definieren.

### 5.9.1 Syntax in EBNF

Eine Beschreibung der EBNF ist in der Wikipedia verfügbar.

---

<code>dyn_font_expr</code>	=	<code>'DynStr(', func, ')';</code>
<code>func</code>	=	<code>strarg   strfmt   expr;</code>
<code>strarg</code>	=	<code>'FahrplanVars::', var_chars   'WetterVars::', var_chars   'FahrplanDaten::NextHalt(", halt_chars, ")'   'FahrplanDaten::LastHalt'   'Str::', var_chars;</code>
<code>strfmt</code>	=	<code>'Fmt(", fmt_chars, ', ', func_args, ')';</code>
<code>func_args</code>	=	<code>func_args, ', ', func   func;</code>
<code>expr</code>	=	<code>'Expr(', logic_expr , ')';</code>
<code>logic_expr</code>	=	Ausdruck dynamische Sichtbarkeitssteuerung
<code>var_chars</code>	=	Gültiger Variablennamen
<code>fmt_chars</code>	=	Gültiger printf Format-String
<code>halt_chars</code>	=	Gültiger Haltname, Etwaige Anführungszeichen " im Haltnamen müssen durch \ " ersetzt werden

---

Tabelle 5.2: Syntax der Dynamischen Schriften in EBNF

### 5.9.2 Erklärung

Die einfachsten dynamischen Schriften greifen schlicht auf Wetter- oder Fahrplanvariablen zu. Dies geschieht mit der gleichen Syntax wie in Logischen Ausdrücken.

Möchte man eine Zahl darstellen, kann man dies mittels der **Expr** Funktion erledigen. Als Argument kann dieser Funktion jeder gültige Logische Ausdruck übergeben werden. Das numerische Ergebnis des logischen Ausdrucks, wird in einen String im Dezimalsystem umgewandelt.

Die komplexeste Art von dynamischen Schriften beinhalten den Aufruf der Funktion **Fmt**. Diese ermöglicht es, einen komplexen Ergebnisstring zu erstellen. Als erstes Argument muss dieser Funktion ein Format-String übergeben werden, welcher bestimmt wie das Ergebnis zusammengesetzt wird. Dabei können die Formatierungsoptionen von printf verwendet werden. Danach werden die Argumente für die Formatierungsplatzhalter übergeben. Es kann hier entweder ein Zugriff auf eine Variable erfolgen, ein logischer Ausdruck in der Form **Expr()** angegeben werden oder wiederum eine Fmt Funktion verwendet werden.

Zur Verdeutlichung einige Beispiele für mögliche Formatanweisungen. Leerzeichen wurden dabei durch `␣` ersetzt. Es wird angenommen, dass *FahrplanVars::h* im Fahrplan als *halt* definiert ist.

Ähnlich wie bei der Sichtbarkeitssteuerung können in dynamischen



Schriften Variablen in der Form *Str::xxx* verwendet werden. Diese können dann über die Objekteigenschaften im Streckeneditor gesetzt werden. Beim Setzen solcher Eigenschaften kann entweder eine fixe Zeichenkette oder wiederum eine dynamische Schrift in der Form *DynStr(...)* verwendet werden. Wird die Variable in einem verschachtelten logischen Ausdruck verwendet, können die Eigenschaften in der gleichen Art wie bei der Sichtbarkeitssteuerung gesetzt werden. ab 2.9

Ausdruck	Ergebnis
<code>DynStr(Fmt("aa%daa", Expr(10)))</code>	aa10aa
<code>DynStr(Fmt("aa%4daa", Expr(10)))</code>	aa_10aa
<code>DynStr(Fmt("aa%4daa", Expr(1000)))</code>	aa1000aa
<code>DynStr(Fmt("aa%-4daa", Expr(1)))</code>	aa1_1aa
<code>DynStr(Fmt("aa%-4daa", Expr(1234)))</code>	aa1234aa
<code>DynStr(Fmt("aa%04daa", Expr(2)))</code>	aa0002aa
<code>DynStr(Fmt("aa%04daa", Expr(11111)))</code>	aa11111aa
<code>DynStr(Fmt("aa%-10saa", FahrplanVars::h))</code>	aahalt_10aa
<code>DynStr(Fmt("aa%8saa", FahrplanVars::h))</code>	aa_8aa

Tabelle 5.3: Beispiele für Formatanweisungen

`FahrplanDaten::NextHalt()` liefert den nächsten Halt *nach* halt. Kommt in halt ein Anführungszeichen " vor, muss dieses durch \ " ersetzt werden.

`FahrplanDaten::LastHalt` liefert den letzten Halt der keine Betriebsstelle ist

### 5.9.3 Performance

Schriften die sich sehr oft ändern, sollten mit Bedacht eingesetzt werden, da sie die fps relativ stark beeinflussen.

Verwendet eine Schrift nur konstante Werte (FahrplanVars oder konstante logische Ausdrücke) so hat diese kaum einen Einfluss auf die Performance

### 5.9.4 Beispiele

```
DynStr(Fmt("Abfahrt in %d Minuten",
Expr((Funktionen::TimeDif(
```

```

FahrplanDaten::Abfahrtszeit("h0"),
Simulation::ZeitInSekunden)) / 60))

DynStr(Fmt("%02d:%02d:%02d",
Expr(Simulation::ZeitInSekunden / 60 / 60),
Expr(Simulation::ZeitInSekunden / 60 % 60),
Expr(Simulation::ZeitInSekunden % 60)))

DynStr(IntToStr(Simulation::ZeitInSekunden * 10))

DynStr(FahrplanVars::Zugziel)

```

---

## 6 Strecken

---

### 6.1 Gleis

#### 6.1.1 Gleiseigenschaften

##### Signal

**Option Gruppensignal** Wenn diese Option aktiviert ist, wird nicht nur **nach** dem Signal nach der tiefsten Geschwindigkeitslimite gesucht, die am Signal gezeigt werden soll, sondern auch **vor** dem Signal. Dabei gilt folgende Regel: Es wird im Bereich +/- 1000 m ab Signal gesucht. Innerhalb dieses Bereichs wird die Suche zusätzlich begrenzt durch das nächst stehende Hauptsignal oder einen Haltepunkt. Wird keine Limite gefunden, wird die aktuell gültige Limite verwendet.

Einrichten eines Gruppenausfahrsignals Neues Signal definieren nach letzter Weiche auf Ausfahrseite (muss an Gleis liegen, welches auf die Strecke führt). Häckchen setzen bei Option "Gruppensignal". Haltepunkte eintragen in den Bahnhofgleisen. In jedem Bahnhofgleis Geschwindigkeitslimite eintragen, die am Gruppenausfahrsignal angezeigt werden soll (z.B. Gleis 1 (Nebengleis) 40 km/h, Gleis 2 (Durchfahrgleis) 120 km/h, Gleis 3 (Überholgleis) 60 km/h). Die Limiten müssen nach dem jeweiligen Haltepunkt und vor dem Gruppenausfahrsignal eingetragen werden (z.B. Position Haltepunkt: 2500, Position Limite: 2501, Position Gruppenausfahrsignal: 2700).

Tipp: Ist die Bahnhofeinfahrtsgeschwindigkeit verschieden von der Ausfahrtsgeschwindigkeit, kann das Einfahrsignal ebenfalls als Gruppensignal definiert werden und Limiten vor dem Haltepunkt für die jeweiligen Gleise gesetzt werden. Dadurch wird erreicht, dass nur Limiten bis zum Haltepunkt gesucht werden und nicht bis zum nächsten Signal, wie es der Fall wäre ohne Option "Gruppensignal".

##### Signaloptionen

**Gleisabhängiges Zusatzsignal** Diese Funktion kann für Zusatzsignale benutzt werden, die je nach Gleis ein bestimmtes Font anzeigen sollen, ähnlich dem Zs2. Bei diesem Zusatzsignal wird aber auf der KBS-Strecke in Fahrrichtung rückwärts nach einem Eintrag für dieses Zusatzsignal gesucht. Mit dem Setzen eines Referenzzeichens (Buchstabe, Zahl) kann der entsprechende Font angesteuert werden. Es wird ab Signal maximal 1000 m rückwärts nach der Zusatzsignalreferenz gesucht.

Einrichten eines solchen Zusatzsignals:

- In einem Gruppenobjekt den entsprechenden Schriftzug wählen und beim Popup-Menü Text aus Eigenschaft der Gruppe den Eintrag "GlabhZSigäuswählen.
- In den Eigenschaften einer Strecke an der gewünschten Position eine neue Signalooption definieren und bei "Gleisabhängiges Zusatzsignal" das Häkchen setzen. Beim Eintrag Referenzzeichen ein Zeichen (Buchstabe, Zahl) eintragen, das dem gewünschten Fontzeichen entspricht.

## 6.2 Streckenobjekte

### 6.2.1 Wiederholung an Achse

ab 2.9

Über den Button *Achse Wiederholung...* besteht die Möglichkeit, Objekte an einer benutzerdefinierten Achse zu wiederholen. Dies macht nur Sinn, wenn bei *Anzahl* mehr als ein Objekt eingetragen ist. Wird die Funktion aktiviert, werden die Objekte nicht entlang des Streckenverlaufs wiederholt, sondern entlang der benutzerdefinierten Achse. Die Achse wird dabei durch einen x/y/z Winkel definiert, der relativ zum aktuellen Gleis verwendet wird.

Die Funktion kann beispielsweise verwendet werden, um einfach eine Baumreihe die in 90° von der Strecke weg verläuft, zu erstellen.

---

## 7 Loks

---

### 7.1 Führerstand

Loksim unterstützt derzeit nur 2D-Führerstände. Der Führerstand besteht aus einer einzelnen Textur auf welche die Instrumente und Schalter gezeichnet werden

Bitmap : Führerstandsbild

Bitmap Nacht : Führerstandsbild, welches in der Nacht bzw. bei Dunkelheit angezeigt wird. Tag- und Nachtführerstandsbild werden dabei linear überblendet um einen fließenden Übergang zu erreichen. Dafür wird im Normalfall die aktuelle (Wetter)helligkeit herangezogen. Ist in der Strecke jedoch eine spezielle Beleuchtung des Gleises eingetragen, werden diese Beleuchtungseinstellungen verwendet.

Es ist zu beachten, dass sich die Instrumente bei Nacht- und Tagführerstand exakt an den gleichen Positionen befinden müssen und die Bitmaps exakt gleich groß sein müssen. Außerdem muss die gleiche Transparenzfarbe verwendet werden bzw. bei beiden Bitmaps der Alphakanal verwendet werden.  
ab 2.8.3

Streckenfenster : Hier müssen die Koordinaten bzw. die Größe des Streckenfensters eingetragen werden. In diesem Bereich wird die 3D-Landschaft hineingezeichnet.

Transparentfarbe : Falls *Transparenz aus Alphakanal* nicht gewählt wird, werden alle Pixel welche die hier angegebene Farbe besitzen vollständig transparent dargestellt.

Transparenz aus Alphakanal : Wird diese Option gewählt, wird die Transparenz aus dem Alphakanal des Bitmaps genommen. Damit können auch halbtransparente Bereiche definiert werden. ab 2.8.3

Auflösung : Größe des Führerstandsbild

### 7.2 Dialog Motor (Antriebsdaten)

Wenn im Folgenden von Motor die Rede ist, so ist das Antriebskonzept im Loksim gemeint. Es wird immer mit den Summen aller Motoren eines Triebfahrzeuges gerechnet. Loksim kennt derzeit keine Differenzierung einzelner Motoren, Motortypen oder Triebfahrzeugarten.

Die nachstehenden Angaben sind zwingend notwendig, damit Loksim mit seinem Antriebsmodell rechnen kann. Leider sind sie nicht immer leicht herauszufinden. Am meisten Infos findet man in den Handbüchern zu den Triebfahrzeugen, wenn dies nicht möglich ist, vielleicht mit einer Suche im Internet.

#### 7.2.1 maximale Geschwindigkeit (km/h)

Wird zur Berechnung des Motorwiderstandes benötigt. Ist gewöhnlich bei jeder Lok irgendwo angeschrieben.

#### 7.2.2 maximale Motorspannung (V)

Wird benötigt zur Berechnung des Motorwiderstandes sowie der je nach Fahrstufe am Motor anliegenden Spannung. Dieser Wert dürfte meist nur Fachleuten bekannt sein. Allgemein kann man sagen, dass er sich im Bereich von 400-700 Volt bewegt mit einem mittleren Wert bei ca. 500-550 Volt. Vor allem moderne Motoren arbeiten aber auch mit Spannungen bis über 1 kV.

#### 7.2.3 Wirkungsgrad der Motoren (%)

Verwendung siehe maximale Lokleistung. Noch ein Wert, der wohl nicht mal immer den Fachleuten bekannt ist. Allgemein zwischen 80 – 95 %. Mit diesem Wert werden alle Verluste repräsentiert, die anfallen zwischen der Leistung, die in die Motoren gesteckt wird und der Leistung, die am Rad zur Verfügung steht.

#### 7.2.4 maximale Lokleistung (kW)

Es handelt sich um die mechanische Leistung, die am Rad anliegt. Bei der Berechnung des Motorstroms wird im Loksim der Verlust dazugerechnet durch Einbezug des Motor-Wirkungsgrades und mit diesem Strom wird dann die Leistung gerechnet, die in den Motor gesteckt wird. Dieser Wert dürfte bei neueren Loks (Asynchronmotoren) oft in Erfahrung zu bringen sein, bei älteren Loks sind jedoch meist die Stunden- und Dauerleistung angegeben, welche immer tiefer als die maximale Leistung sind. Wenn die Loks eine Primärstrombegrenzung haben, kann über die Primärleistung  $Oberspannung[V] * \frac{Primärstrom[A]}{1000}$  ebenfalls auf die ungefähre maximale Leistung geschlossen werden. Die %-Zahlen sind Näherungswerte. Es ist mit einer Streuung von ungefähr +/- 10% zu rechnen. Der Bezug auf die Primärleistung ist am wenigsten genau. Die maximale Lokleistung ist

- ca. 145 % der Dauerleistung am Rad
- ca. 135 % der Dauerleistung an der Motorwelle
- ca. 135 % der Stundenleistung am Rad
- ca. 125 % der Stundenleistung an der Motorwelle

- ca. 80 % der Primärleistung

Bei älteren Loks wird noch die alte Einheit PS verwendet. 1 PS entspricht 736 W oder 0.736 kW.

#### 7.2.5 Maximaler Motorstrom (A)

Wird benutzt, um die Fahrleistung zu begrenzen auf die maximal zulässigen Motorstromwerte. Ist bei älteren Loks meist irgendwo im Führerstand angeschrieben und in den Handbüchern verzeichnet, bei neueren selten bekannt. Wird die Loksim-Funktion *Motorstrom berechnen* benutzt, berechnet Loksim den Motorstrom aus max. Leistung und max. Motorspannung. Dies entspricht aber vor allem bei älteren Loks oft nicht dem max. Motorstrom, er ist meist höher.

#### 7.2.6 Wirkungsgrad Transformator (%)

Verwendung siehe Oberspannung. Auch diese Angabe ist kaum in Erfahrung zu bringen. Der Wirkungsgrad liegt im Bereich von 90 – 97 %.

#### 7.2.7 Oberspannung (V)

Aus momentaner Leistung und Wirkungsgrad des Transformators wird der Oberstrom berechnet. Die benutzte Netzspannung ist wiederum bekannt.

#### 7.2.8 maximale Zugkraft (kN)

Wird unter anderem zur Berechnung der Beschleunigung verwendet. Dieser Wert dürfte meist irgendwo in der Literatur/Internet zu finden sein. Bei älteren Loks wird die Zugkraft noch in kg oder kp angegeben. 1 kg oder 1 kp = 9,81 N oder 0,00981 kN

#### 7.2.9 Motorparameter

Die Erklärungen und Anleitungen zu den Motorparametern gelten vor allem für elektrische Triebfahrzeuge, die mit Stufenschalter oder ähnlich ausgerüstet sind. Für moderne Drehstromloks, Dieselloks mit Gangschaltung und Dampfloks ist es derzeit nicht möglich, ein vorbildgerechtes Verhalten aufgrund von lokspezifischen Daten einstellen zu können. Trotzdem kann ein annähernd vorbildliches Verhalten durch geschicktes Einstellen der Parameter erreicht werden.

Mit Hilfe der drei Motorparameter wird eine hyperbelähnliche Kurve berechnet, die als Widerstandskurve im Dialog ersichtlich ist und als Basis zur Simulation des Antriebsverhaltens des Triebfahrzeugs dient. Diese Kurve ist typisch für Triebfahrzeuge mit Reihenschlussmotoren (auch Serie- oder Universalmotoren genannt). Diese Motoren wurden verwendet bis zum Aufkommen moderner

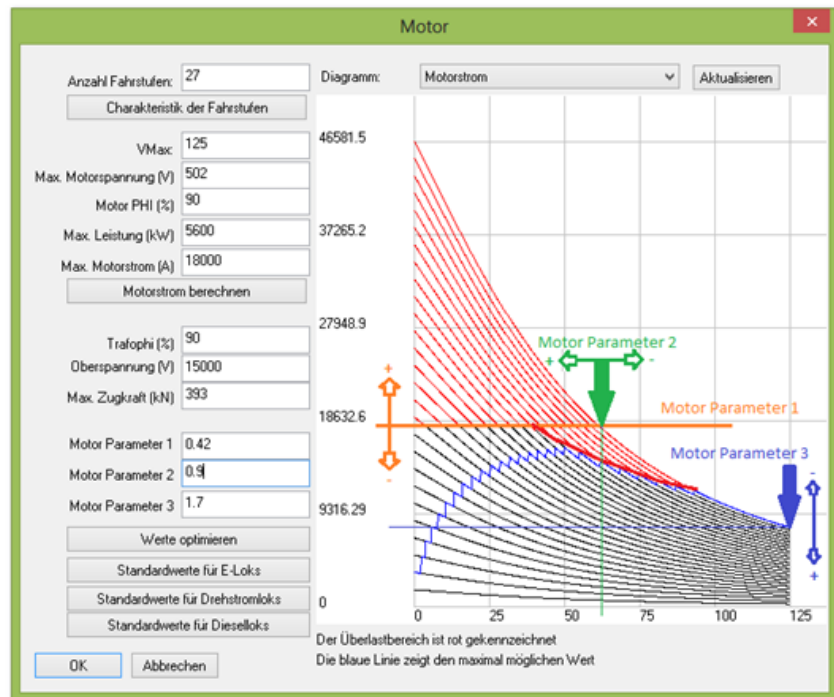


Abbildung 7.1: Motorparameter der Ae 610 der SBB

Loks mit Asynchron-Drehstrommotoren, die eine andere Kennlinie besitzen.

Zur Veranschaulichung, wie die drei Parameter (am Beispiel der Ae 610 der SBB) eingestellt werden, siehe Abbildung 7.1

#### Motorparameter 1 (orange)

Dient zur Festlegung der Stufe (oder Punkt zwischen zwei Stufen), bei der der maximale Motorstrom erreicht wird. (Im Beispiel unten zwischen der 10. und 11. Stufe). Dazu ganz links bei Geschwindigkeit 0 aufsteigend die Anzahl schwarzer Linien zählen. Parameter 1 so lange verändern, bis gewünschte Stufe erreicht wird.

#### Motorparameter 2 (grün)

Legt fest, ab welchem Punkt der Motorstrom auf der höchsten Fahrstufe in Bezug auf eine bestimmte Geschwindigkeit zu sinken beginnt. Durch Verändern des Parameters 2 kann der Punkt, an dem sich die waagrechte Linie des max. Motorstroms und die Linie der höchsten Stufe kreuzen, so waagrecht verschoben werden, bis die Senkrechte durch diesen Punkt unten auf der Geschwindigkeitsskala auf die gewünschte Geschwindigkeit trifft.



### Motorparameter 3 (blau)

Bestimmt den Motorstrom, der bei maximaler Geschwindigkeit auf der höchsten Fahrstufe noch fließt. Mit dem Verändern des Parameters 3 kann die Höhe des Motorstroms eingestellt werden, der bei max. Geschwindigkeit noch fließt.

Es ist zu beachten, dass sich die Motorparameter 2 und 3 gegenseitig beeinflussen. Es ist deshalb meist nötig, den nicht veränderten Parameter etwas nachzuregeln, bis die Kennlinien mit dem Vorbild übereinstimmen.

Auf Abbildung 7.1 auf der vorherigen Seite wurde die Oberstrom- (oder Primärstrom-)begrenzung durch eine rote Linie gekennzeichnet. Bei Loks, die keine Oberstrombegrenzung haben, fällt diese "Delle" weg.

### Werte optimieren

Wenn detaillierte Werte einer Lok bekannt sind, kann man diese in eine Tabelle eingeben und Loksim erstellt daraus eine Widerstandskurve, indem es selbst die drei Parameter berechnet.

### 7.2.10 Standardwerte für E-Loks, Drehstromloks und Dieselloks

Stehen keine genauen Angaben für eine Lok zur Verfügung, kann man mit dieser Funktion die Kennlinien durch Loksim erstellen lassen. Ebenso dient sie als Ausgangsbasis und der Ersteller des Führerstandes kann durch Veränderung der Motorparameter die Kennlinien nachträglich manuell nach seinen Wünschen einstellen (siehe oben).

### Drehstromloks

Der Parameter 1 hat bei diesen Loks eigentlich keine Bedeutung, denn hier wird mit dem Fahrhebel die prozentuale Zugkraft eingestellt. Die „Stufenlinien“ müssten daher eigentlich waagrecht liegen. Loksim ist aber zur Zeit noch nicht für Drehstromloks optimiert. Immerhin kann mit dem Parameter 1 der Punkt mitbeeinflusst werden, an dem die maximale Lokleistung zur Verfügung steht (siehe auch Parameter 2).

Annäherungsweise kann der Parameter 2 so eingestellt werden, dass die maximale Leistung bei der Geschwindigkeit erreicht wird, bei der die Zugkraft zu sinken beginnt.

Ebenso kann mit dem Parameter 3 der Punkt angesteuert werden, an dem noch eine bestimmte Zugkraft bei max. Geschwindigkeit erbracht wird.

## Dieselloks

Wenn es sich um Diesel-Generator-Loks handelt, kann die Lok wie eine Stufenschalterlok eingestellt werden. Die Dieselthematik ist zur Zeit im Loksim noch nicht umgesetzt.

## Dampfloks

Da die Dampfloks ein sehr ähnliches Verhalten wie Stufenschalterloks haben, kann ihr Verhalten mittels der drei Parameter ziemlich vorbildlich nachgebaut werden, allerdings mit „artfremden“ Parametern (Spannung, Strom). Die Dampfthematik ist derzeit im Loksim noch nicht umgesetzt.

### 7.2.11 Diagramm

Mit dem Popup-Menü kann zwischen den Ansichten Leistung, Oberstrom, Motorstrom, Motorspannung, Widerstand und Zugkraft gewählt werden.

### 7.2.12 Bemerkung

Die Zugkraft ist mit einem Fehler behaftet, was sich insbesondere bei Drehstromloks mit Zugfahrkrafthebeln bemerkbar macht. Das Antriebsmodell des Loksim wird derzeit überarbeitet (inklusive Bremsen) und voraussichtlich in einer der nächsten Versionen veröffentlicht.

## 7.3 Indusi/PZB

In Version 2.8.2 wurde eine Überarbeitung der PZB90 eingebaut. Dieser Abschnitt bezieht sich derzeit, nur auf die PZB-Versionen, welche von dieser Überarbeitung betroffen sind.

### 7.3.1 Art der Indusi

Folgende PZB-Arten wurden mit Version 2.8.2 erneuert:

- PZB90 I60R
- PZB90 I60/ER24
- PZB90 PZ80R
- PZB90 I80

Die PZB Version kann für diese Typen im PZB Einstellungsdialog geändert werden

Eine detaillierte Beschreibung der Systeme kann im betrieblich-technischen Regelwerk der DB Netz AG nachgelesen werden.

Bei I60R ertönt der Sound für die Freitaste nur, wenn tatsächlich eine Befreiung möglich ist. Bei I60/ER24 ertönt dieser Sound immer wenn die Freitaste gedrückt ist.

Kontrolllauf und PZ80R Kontrollschalter sind derzeit nicht implementiert.

Jene Optionen die mit (*veraltet*) gekennzeichnet sind, sollten bei neuen Loks nicht mehr verwendet werden. Diese Optionen stehen nur aufgrund Rückwärtskompatibilität zu älteren Führerständen zur Verfügung

### 7.3.2 Sounds

Indusihupe (WT, FT) : Sound welcher bei Betätigung von Wachsamkeitstaste bzw. Freitaste abgespielt wird. Dieser wird in einer Schleife abgespielt, solange die entsprechende Taste gedrückt ist.

Indusibefehl : Sound welcher in einer Schleife abgespielt wird, solange die Befehlstaste aktiv ist.

Zwangsbr.-Indusi : Dieser Sound wird in einer Schleife während einer Zwangsbremmung abgespielt.

Zwangsbr.-Indusi nur einmal : Dieser Sound wird exakt einmal am Anfang einer Zwangsbremmung abgespielt.

Ende 500Hz Überwachung : Sound wird am Ende einer 500Hz Überwachung abgespielt, falls sie im restriktiven Modus endet.

Überschreiten V-Pruef : Dieser Sound wird beim Überschreiten der PZB-Prüfgeschwindigkeit wiederholt abgespielt. Je nach PZB-Art kommt dieser Sound sofort bei Überschreiten oder erst nach einer definierten Zeitspanne.

### 7.3.3 Leuchtmelder und Anzeigen

Die Leuchtmelder und Anzeigen werden wie andere Anzeigen im Lokeditor definiert. Bei Einsatz von PZ80R sollte von den Leuchtmeldern nur der Leuchtmelder Indusi 95 (LVZ grün) definiert werden, aber dafür die Anzeige IndusiVZiel. Bei den anderen Arten sollte alle Leuchtmelder gesetzt werden, jedoch die Anzeige IndusiVZiel nicht.

### 7.3.4 PZB Einstellungen

Im Lokeditor ist über das Menü Bearbeiten - PZB Einstellungen ein Dialog abrufbar, der weitere Einstellungen für die PZB ermöglicht.

Zum Großteil geht es hier um Eigenschaften der PZB die nicht exakt aus den uns zur Verfügung stehenden Unterlagen implementiert werden konnten oder wo es in der Praxis konträre Erfahrungen gibt. Jedoch gibt es auch Einstellungen die bekanntermaßen von Lok zu Lok unterschiedlich sind

Version : Hier kann die Version der PZB90 ausgewählt werden. Es stehen alle Versionen die in der Realität verwendet werden und

wurden (1.5, 1.6 und 2.0) zur Verfügung

- Befehlstaste ist ein Schalter : Ein Druck auf die Befehlstaste aktiviert die Befehlstaste und ein zweiter Druck deaktiviert diese.
- Befehlstaste ist ein Taster : Befehlstaste ist nur aktiv, solange die Taste gedrückt wird. Dies ist meist bei neueren Tzf der Fall.
- Zeit für Wachsamkeitstaste : Definiert die Zeit in Millisekunden innerhalb welcher nach einer 1000Hz Beeinflussung die Wachsamkeitstaste gedrückt werden muss. Normalerweise sind dies 4 Sekunden, im Fall der MVB jedoch 2,5 Sekunden.
- Befreiung aus Zwangsbremse... : Annahme: Bei Stillstand nach einer Zwangsbremse aufgrund einer 1000Hz Überwachung sind bereits 700m oder mehr ab Beginn der Beeinflussung vergangen. Nun muss die Freitaste zur Befreiung aus der ZB betätigt werden.
- Falls die Option kann gleichzeitig Befreiung aus 1000Hz Überwachung bewirken gesetzt ist, bewirkt ein Druck auf die Freitaste die Befreiung aus der ZB, sowie die Befreiung aus der 1000Hz Überwachung.
- Ist hingegen bewirkt niemals gleichzeitig Befreiung aus einer Überwachung gewählt, bewirkt ein Drücken der Freitaste ausschließlich die Befreiung aus der ZB.
- Dauerbetätigung PZB Tasten : Wird hier eine Option gesetzt, wird die entsprechende Taste nach der bestimmten Zeit und/oder Entfernung unwirksam. 0 bedeutet, dass die Taste kein Zeit- bzw. Entfernungsmaximum besitzt.
- Die Richtlinien der ÖBB geben eine maximale Distanz von 225m für alle Tasten an.
- Status 'befreit' wird an überlagerte Überwachung weitergegeben : Wird diese Option gesetzt, wird der Status 'befreit' einer 1000Hz Überwachung an eine überlagerte 1000Hz Überwachung weitergegeben. Angenommen folgende Situation: 0m/1000Hz, 800m/Befreiung, 1000m/1000Hz, 1400m/500Hz: Ist diese Option nicht gesetzt, erfolgt am 500Hz Magnet keine Zwangsbremse, solange man die entsprechende Geschwindigkeit einhält. Ist die Option gesetzt, bekommt man am 500Hz Magnet immer eine ZB aufgrund nicht erlaubter Befreiung
- Sonderform : Über diese Einstellungen können Sonderformen der PZB simuliert werden.
- Derzeit ist nur die Sonderform *Stadtbahn* möglich. Bei dieser Variante gibt es neben dem Wechselblinker ein Gleichblinker, bei welchem 65km/h gefahren werden dürfen.

## 7.4 Türsteuerung

mind. Türschliesszeit (Sek) : Geben Sie hier an, wieviele Sekunden die Türen mind. zum Schließen benötigen. Durch div. Einflüsse kann das Schließen auf ggf. zufällig länger dauern.

Deaktiver Türmelder (Gtz) : Geben Sie hier an, welchen Melderzustand bei Güterzugdienst verwendet werden soll. Sie können zwischen Zustand ein bzw. aus wählen.

Türsnd (schliessen) : Geben Sie hier die Sounddatei an, die während des Schließen der Türen abgespielt werden soll. Das Geräusch darf nur ein Dauerton beinhalten.

Türsnd (geschlossen) : Geben Sie hier die Sounddatei an, die nach dem Schließen der Türen abgespielt werden soll. Das Geräusch darf nur ein Dauerton beinhalten.

Türsnd (öffnen) : Geben Sie hier die Sounddatei an, die während des Öffnen der Türen abgespielt werden soll.

Für ein optimales Zusammenspiel von Sound und Anzeigen im Führerstand, sollte die *mind. Türschliesszeit* der Dauer der wav-Datei *TürSnd (schließen)* entsprechen.

## 7.5 Diverses

### 7.5.1 Wegmessung / Zuglängenzähler

ab 2.8.3

Der Zuglängenzähler besteht im Loksim aus zwei Sounds:

Sound Wegmessung Beginn : Dieser Sound wird beim Start der Wegmessung abgespielt (optional)

Sound Wegmessung Ende : Dieser Sound wird nachdem eine Zuglänge zurückgelegt wurde abgespielt

Wegmessung aktivierbar ab x km/h : Falls diese Einstellung auf einen Wert größer als 0 eingestellt ist, lässt sich die Wegmessung erst ab der eingestellten Geschwindigkeit starten und nicht im Stillstand

---

## 8 Wettereditor

---

### 8.1 Niederschlag

Ab Version 2.8.2 besteht in Wetterdateien die Möglichkeit den Schneefall zu steuern.

*Die Umsetzung des Niederschlags im Loxsim ist derzeit auf keinen Fall ausgereift und sieht nicht in allen Situationen wirklich gut aus. Er wurde ursprünglich als "Weihnachtsüberraschung" eingebaut und noch nicht vollständig umgesetzt.*

Schneefall kann im Wettereditor pro Zeitbereich mit der Checkbox Schneefall aktiviert werden

---

## 9 Allgemeines

---

### 9.1 Dateien überschreiben

ab 2.9

Im Gegensatz zu vielen anderen (3D-)Anwendungen verwendet Loksim ein vollkommen offenes Dateiformat für sämtliche Addons. Zusätzlich sind Dateien nicht speziell an ein Package gebunden, sondern jedes einzelne Objekt, Textur, Führerstand, etc. kann im LoksimEdit geöffnet, betrachtet und sogar geändert werden.

Je nach *Lizenz des Package* und der darin verwendeten Teile können Teile eines anderen Package auch - eventuell sogar in einer abgeänderten Form - in einem eigenen Package wiederverwendet werden.

Bei einer Änderung einer fremden Datei ist jedoch *immer eine Kopie anzulegen*, fremde Dateien dürfen *niemals überschrieben* werden.

Lange Zeit wurde diese essentielle Übereinkunft technisch überhaupt nicht überprüft und nur eine sorgfältige Verwendung des LoksimEdit hat vor unabsichtlichem Überschreiben fremder Dateien geschützt. Mit Version 2.9 wurde hierfür zusätzlich eine technische Unterstützung eingeführt, die das unabsichtliche Überschreiben fremder Dateien verhindern soll. Die Implementierung basiert dabei auf folgenden Grundsätzen:

- Jede Datei besitzt einen oder mehrere Autoren. Diese können im LoksimEdit über *Datei - Eigenschaften* eingetragen werden. Mehrere Autoren können dabei mit einem ; (Strichpunkt) getrennt eingetragen werden. Es wird davon ausgegangen, dass nur eine Person welche als Dateiautor eingetragen ist, die Datei editieren sollte.
- Im LoksimEdit ist der eigene Name (oder ein in der "Loksim-Welt" verwendetes Pseudonym) unter *Datei - Editor Optionen - Sonstige - Standard Ersteller* eingetragen.
- Wird eine Datei gespeichert ohne, dass ein Standard Ersteller in den Optionen eingetragen ist, wird eine Warnung angezeigt (Abbildung 9.1 auf der nächsten Seite)
- Es wird eine Warnung angezeigt, falls versucht wird eine Datei ohne gesetztem Dateiautor zu speichern (Abbildung 9.1 auf der nächsten Seite)

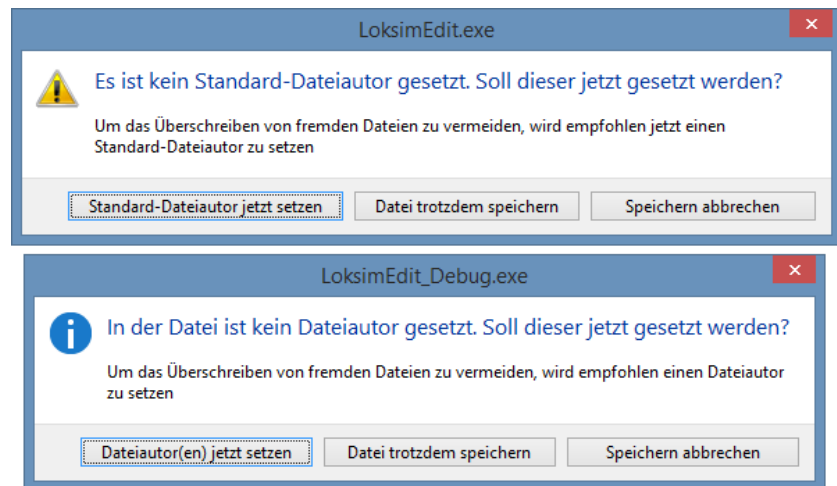


Abbildung 9.1: Fehlermeldungen beim Speichern von Dateien ohne eingetragenen Standardersteller (oben) bzw. ohne eingetragenen Dateiautor (unten)

- Es wird gewarnt wenn versucht wird eine Datei zu speichern, bei welcher man nicht selbst als Dateiautor eingetragen ist (Abbildung 9.2 auf der nächsten Seite)
- Falls beim Speichern einer Datei eine bestehende Datei überschrieben wird, wird überprüft ob bei der bestehenden Datei der Standard Ersteller als Dateiautor eingetragen ist. Ist er dies nicht, wird wiederum eine Warnung angezeigt (Abbildung 9.2 auf der nächsten Seite)

Diese Lösung bietet keinen vollständigen Schutz vor dem Überschreiben fremder Dateien und schützt klarerweise überhaupt nicht vor dem Überschreiben von Dateien außerhalb des Loks>Edit. Ein vollständiger Schutz vor unbeabsichtigten Änderungen wäre wünschenswert, ist jedoch bei dem offenen Konzept von Loks'im nicht einfach umzusetzen. Allerdings sollten die in Version 2.9 eingeführten Warnungen zumindest einen Teil der unabsichtlichen Änderungen verhindern.

## 9.2 Package erzeugen

Ein Package kann über das Menü Datei - Package erzeugen... erstellt werden.

Wird beim Suchen der abhängigen Dateien erkannt, dass eine Datei fehlt wird am Ende des Vorgangs eine Liste mit den fehlenden Dateien gezeigt. ab 2.9



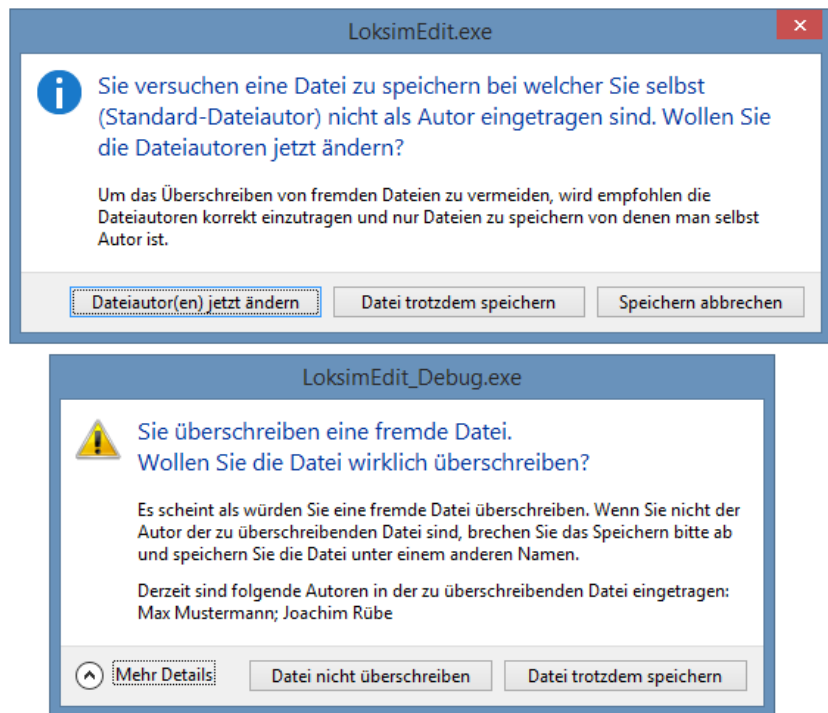


Abbildung 9.2: Fehlermeldungen beim Speichern von Dateien bei denen man nicht selbst Dateiautor ist (oben) bzw. beim Überschreiben von fremden Dateien (unten)

### 9.2.1 Diese Dateien löschen / Packages deinstallieren

Über die Schaltfläche Datei / Package hinzufügen lassen sich Dateien hinzufügen, die bei der Installation des Package gelöscht werden sollen. Hierbei ist es wichtig, dass nur Dateien hinzugefügt werden dürfen, die sicher nicht von anderen Packages benutzt werden.

Seit Version 2.8.2 können auch ganze Packages hinzugefügt werden. Wird ein Package hinzugefügt, wird dieses Package beim Benutzer deinstalliert. Die Identifikation des Package erfolgt dabei über eine Prüfsumme und nicht über den Dateinamen des Package. Ändert man etwas am Inhalt des Package, ändert sich auch dessen Prüfsumme.

Es wird empfohlen bei neuen Versionen einer Strecke oder Führerstand die alten Versionen des Package in diese Package deinstallieren Liste hinzuzufügen. So werden beim Benutzer stets die nicht mehr benötigten Dateien gelöscht

### 9.2.2 l3dmeta Dateien

ab 2.9.6

Zur Definition von Eigenschaften von Dateien die nicht im Loksime XML-Format abgespeichert werden (wie zum Beispiel Texturen oder 3D-Objekte in externen Formaten) können .l3dmeta Dateien

angelegt werden, die beim Erzeugen von Packages ausgewertet werden.

Beispiel: In einem Package für eine Loksim-Strecke wird das Objekt `\Objekte\Bahnhof\MaxM\BhfX.dae` verwendet. Der Autor *Max Mustermann* möchte explizit ausdrücken, dass dieses Objekt von ihm erstellt wurde und unter der Lizenz *Loksim-AddonLizenz-2019-08* steht.

Um dies zu erreichen, ist es möglich eine Datei `\Objekte\Bahnhof\MaxM\BhfX.dae.l3dmeta` mit folgendem Inhalt zu erstellen:

```
<?xml version="1.0"?>
<METADATA>
  <Props FileAuthor="Max Mustermann"
    FileLicense="\Lizenzen\Loksim-AddonLizenz-2019-08.pdf" />
</METADATA>
```

Wird nun ein Package erzeugt welches die Datei `\Objekte\Bahnhof\MaxM\BhfX.dae`, wird automatisch auch die Datei `\Objekte\Bahnhof\MaxM\BhfX.dae.l3dmeta` und `\Lizenzen\Loksim-AddonLizenz-2019-08.pdf` ins Package mit aufgenommen. Somit ist stets nachvollziehbar, unter welcher Lizenz das Objekt steht und wer der Autor dieser Datei ist.

### 9.3 Automatische Sicherung von Dateien

Der Editor speichert regelmäßig Sicherungskopien der gerade bearbeiteten Datei(en) im *LastWork* Ordner im Datenverzeichnis. Dies ist nützlich falls man eine Änderung nach ein paar Tagen rückgängig machen möchte oder eine Datei unabsichtlich gelöscht hat.

Diese Sicherungsdaten werden auch für die Integration des sogenannten Restart Manager von Windows verwendet. Dies ermöglicht es nach einem etwaigen Absturz des LoksimEdit die zuletzt geänderten Dateien automatisch zu öffnen. Außerdem wird diese Funktion bei Neustarts des LoksimEdit bzw. von Windows verwendet (zB bei Installation einer neuen Loksim-Version oder bei Windows Updates).

ab 2.9.3

*Hinweis:* Wir empfehlen Addon-Bauern trotzdem unbedingt eine regelmäßige Sicherung des Datenverzeichnis und das Aufheben von mehreren (älteren) Versionsständen.

### 9.4 Logging

ab 2.9.2

Im Verzeichnis `%LOCALAPPDATA%/Loksim3D` werden für Loksim3D und für den LoksimEdit Logs erstellt. Diese umfassen derzeit noch wenige Informationen, werden aber in Zukunft mehr Daten enthalten.

In erster Linie können diese Logs für die Behebung von Fehlern genutzt werden und sie werden deshalb automatisch bei Fehlerberichten mitgesendet. Für Addon-Entwickler ergibt sich jedoch auch die Möglichkeit durch diese Logs (zusätzliche) Fehlermeldungen für die eigenen Addons mitzubekommen. Insbesondere in der Simulation können aus diversen Gründen nicht alle Fehlermeldungen (zB bzgl. fehlenden Dateien) angezeigt werden, wir bemühen uns diese in Zukunft zumindest in den Logs sichtbar zu machen.

---

### III. Anhang

---

---

## 10 Versionshistorie

---

### 10.1 Version 2.10

#### 10.1.1 Neue Funktionen

- Loksim: RailDriver Integration
- LoksimEdit: Überarbeitung Führerstandseditor
- LoksimEdit: "Grab Geste" fuer KBS-, Strecke-, Objektbitmap- und Führerstandsbitmap-View
- LoksimEdit: Letzer Zustand laden im Create Package Dialog

#### 10.1.2 Kleinere Änderungen

- Loksim: SEP um textEBuLa erweitert
- Verbesserte Berechnung der Bounding Box von 3D-Fonts
- LoksimEdit: 2D-View aktualisiert sich wenn Objekte geändert werden
- LoksimEdit: Englische Übersetzung für (neue) Teile des LoksimEdit
- LoksimEdit: Sichtwinkel bei Update in KBS-, und Strecke-View wiederherstellen
- LoksimEdit: Im Führerstandeditor sind veraltete Optionen deaktiviert
- PackageManager: Option Transaktionen zu deaktivieren

#### 10.1.3 Fehlerkorrekturen

- Simulator: Bugfix bei Indusi zusätzlich: Korrektur der Berechnung der Signalposition
- Simulator: Bugfix Fst-Überblendung im Stand
- Simulator: Crash beim Blättern im Fahrtenschreiber
- LoksimEdit: Bugfix (Crash) Speichern von Backups wenn Pfad nicht beschreibbar

### 10.2 Version 2.9.6

22. Oktober 2019

### **10.2.1 Neue Funktionen**

- LoksimEdit: Dateilizenz kann angegeben werden
- LoksimEdit: Beim Erzeugen von Packages werden .l3dmeta Dateien ausgewertet - kann zum Definieren der Lizenz von externen Dateien verwendet werden

### **10.2.2 Kleinere Änderungen**

- Simulator: Anzeige Auflösung Führerstand im Auswahldialog
- LoksimEdit: Meldung wenn Datei-Speichern fehlschlägt

### **10.2.3 Fehlerkorrekturen**

- LoksimEdit: Bugfix beim Bestimmen von Abhängigkeiten von externen Objektformaten beim Erzeugen von Packages

## **10.3 Version 2.9.5**

23. September 2019

### **10.3.1 Neue Funktionen**

- Simulator Exchange Protocols (SEP)

### **10.3.2 Kleinere Änderungen**

- Normalenvektoren haben immer Länge 1
- Verbessertes Transparenzhandling bei Verwendung externer Objektformate
- Umstellung Compiler: Visual C++ 2019

### **10.3.3 Fehlerkorrekturen**

- Editor: Standard Dateinfo und -autor wird nun bei jedem Dateityp korrekt übernommen
- Lagekorrektur der Flügelschienen an der einfachen Weiche
- Bugfix Checkbox Objekt in Sichtweite ändert Sichtbarkeit nicht", wenn Funktion in Signalen verwendet wird

## **10.4 Version 2.9.4**

20. Juli 2018

#### 10.4.1 Neue Funktionen

- Erweiterung der Joystick-Schnittstelle. Es sind mehr als ein Joystick möglich
- Joysticksteuerung (Buttons) erweitert
- Erweiterung der TCP-Schnittstelle, Zustand der LM55/70/85 wird gesondert übertragen

#### 10.4.2 Kleinere Änderungen

- LM Befehl40 im TCP-Protokoll nachgerüstet
- PZB90 Restriktiver Modus wird nun korrekt an TCP, OLE und Fahrtenschreiber gemeldet
- Weichendarstellung verbessert
- Abfrage der aktuellsten Loxsim-Version über HTTPS anstatt HTTP
- Objekteditor: Textkoordinaten für einen Punkt auf einer Fläche für alle Flächen verwenden
- Objekteditor: Beim Drücken von <ALT> werden alle Textkoordinaten auf einmal verschoben

#### 10.4.3 Fehlerkorrekturen

- Kleiner Bugfix bei der DKW-Darstellung
- Bugfix Package Erstellen: Abwählen von Dateien funktioniert wieder
- Bugfix PZB90: LM85/70/55 leuchtet während restriktiver 1000Hz Beeinflussung nicht mehr
- Korrektur der Fahrplandarstellung

### 10.5 Version 2.9.3

1. Jänner 2017

#### 10.5.1 Neue Funktionen

- Neue Optionen für Gleise: Keine 3D-Darstellung, senkrechte Normalenvektoren und neue Transparenzoptionen
- Fahrplanabhängige Sichtweite
- Editor: Objekteditor zeigt Fehler bei Flächen an
- Editor: Integration Restartmanager
- Editor: Neuer Package Erzeugen Dialog und erweiterte Package Informationen

- Editor: Backup von noch nicht gespeicherten Dateien im Last-Work Ordner
- Editor: Neue Option *Ordnerstruktur übernehmen* bei Texturen optimieren
- Objekteditor: Neuer Dialog für Punkte zu Fläche hinzufügen
- Fonteditor: Textur zoombar
- Objekteditor: Normalenvektoren für runde Objekte berechnen

### 10.5.2 Kleinere Änderungen

- FahrplanDaten::LastHalt auch in Sichtbarkeitsausdruecken verfügbar
- paths.ini kann nun auch <Registry> anstatt Pfad enthalten
- Interne Umstellung der Gleiserzeugung
- Indusi I60 ueberwacht nach 1000Hz in Stellung O auf 95km/h (zuvor 85km/h)
- Editor: Bessere Kennzeichnung von fehlerhaften Eingaben in einigen Dialogen
- Editor: LastWork Ordner behält bis zu 500 Dateien auf und sichert auch nichtgespeicherte Dateien
- Editor: Liste von Variablen im Objekteigenschaften-Dialog in Zwischenablage kopierbar
- Streckeneditor: Löschen aller Objekte eines Gleis auf einmal möglich
- Strecken- und KBS-Editor: Anzeige Positionsinformationen über Info-Button

### 10.5.3 Fehlerkorrekturen

- Absturz beim Laden von Objekten die Punkte oder Linien enthalten aus externen Formaten behoben
- Abstürze bei .l3dobj Objekten mit nicht-planaren Flächen behoben
- Verschwinden von Objekten in bestimmten Situationen behoben
- Zulassen von sehr kleiner Streckenrotation
- Bugfix Behandlung von relativen Pfaden in speziellen Fällen
- Bugfix Sichtbarkeit große Objekte



- Simulator: Bei ausgeblendeten Kennziffern wird ein Limit am Hauptsignal dennoch aktiv
- Simulator: Probleme bei LZB-Ende behoben
- Strecken- und KBS-Editor: Absturz bei bestimmten GPAs behoben
- Streckeneditor: Bugfix Eigenschaftsname mit Sonderzeichen (Eigenschaften Gruppenobjekte)
- Streckeneditor: Absturz GPA behoben

## 10.6 Version 2.9.2

11. August 2015

### 10.6.1 Neue Funktionen

- Zufallsdrehung bei Objekten auf Strecke möglich
- Zeichenketten in logischen Ausdrücken
- Editor: Neue Funktion "Texturnutzung optimieren"
- Objekteditor: Neuer Dialog für Punkte zu Fläche hinzufügen

### 10.6.2 Kleinere Änderungen

- Indusi-Einstellung PZ80R wird automatisch auf PZB90 PZ80R umgestellt
- Kommandozeilen-Argument /renderstats:1 zeigt Anzahl von DrawCalls und Triangles
- Optimierung bei externen Objektmodellen wenn kein Alpha-Kanal in der Textur verwendet wird
- Standardmäßiges Erstellen von Logs
- Reihenfolge der Gleise bei Verwendung von verknüpften BÜs nicht mehr relevant
- Editor: FilePicture wird bei Speichern-Unter kopiert
- Editor: Dateien die über "Doppelklick" im LoksimEdit geöffnet werden, erscheinen in RecentFileList
- Editor: Beleuchtung mehr an Simulation angenähert
- Editor: Löschen in Baumansichten mittels "Entf"
- Gruppenobjekteditor: Auswahl nach dem Löschen eines Objekts verbessert
- Fonteditor: Speichern-Unter bei Fonts überschreibt existierende Texturen nicht mehr

- PackageManager: Anzeige von Fehlern bei "Installation rückgängig machen"

### 10.6.3 Fehlerkorrekturen

- Bugfixes bei Sichtbarkeitssteuerung
- Bugfix Rotation Streckenobjekte
- ObjektEditor: Verschwinden von Objekten in seltenen Fällen
- ObjektEditor: Texturkoordinaten "Rückseite einfügen"
- Streckeneditor: Setzen von Inhalt bei Textfeldern ohne Name bzw. bei Textfeldern mit dynamischer Sichtbarkeit
- Streckeneditor: BÜ bei welchem der Name Teil des Namens eines andere BÜs ist, kann nun auch im GUI angelegt werden

## 10.7 Version 2.9.1

11. Jänner 2015

### 10.7.1 Neue Funktionen

- Streckeneditor: (Landschafts)objekte können nach Position sortiert werden

### 10.7.2 Kleinere Änderungen

- Optimierung Option "Alle Texturen beim Start laden"
- Vergrößerung von Eingabefeldern im Editor: Neu-Dialog, Rail - Höhenwerte der Bettung, Strecke - Unterbrechung der Höhenlinie
- Performanceverbesserungen Packageinstallation
- Behandlung nicht planarer Flächen bei Objekten die bis inkl. Version 2.9 erstellt wurden so wie in Version 2.8.3
- Anzeige MessageBox bei fehlendem Sound im Simulator deaktiviert
- Streckeneditor: Laden von Objekten für 2D-Ansicht erfolgt im Hintergrund
- Streckeneditor: Graphikobjekte bei Streckeneigenschaften (Indusi-Magnete, Tafeln, etc) können nun jedes unterstützte Graphikformat sein (nicht nur .l3dgrp)
- Streckeneditor: Button "Rollmaterial anzeigen" in "Rollmaterial ausblenden" umbenannt
- Kursbucheditor: Standardauswahl bei neuer Verbindung im KBS-Editor optimiert

### 10.7.3 Fehlerkorrekturen

- "Höhe über Landschaft" bei global gesetzter Verschiebung in Y-Richtung
- Gleichzeitiger Einsatz von "Objekt mitdrehen" und "Zoomfaktor"
- Texteigenschaft bei bestimmten Fonts
- Streckenobjekte an Achse wiederholen und Verschiebung von Objekten
- Einlesen "2. Länge" bei diversen Instrumenten von alten Führerständen
- Editor: Mehrere 3D-Fenster gleichzeitig bedienen
- Editor: Gehe-Zu-Position Dialog aktualisiert 3D-Ansicht nicht
- Editor: Bugfix Neu-Dialog bei geöffneter 3D-Ansicht
- Editor: Korrektur Anzeige fehlender Dateien beim Erstellen von Packages
- Editor: Bugfix Erstellen eines Package (Versionsinfo, zu löschende Dateien und Packages)
- Objekteditor: Größenänderung Textur wird bei Drücken von "Übernehmen" Button erkannt
- Objekteditor: Flächen kopieren
- Gruppenobjekteditor: Nicht gesetzte Eigenschaft wird standardmäßig zu Wahr "ausgewertet"
- Gruppenobjekteditor: Objekte auf/ab
- Gruppenobjekteditor: Punkte verschieben/zoomen Fonts
- Streckeneditor: "HelligkeitProzent" nicht von manuell eingestellter Helligkeit beeinflusst
- Streckeneditor: Erweiterte Objekteigenschaften bei Signalen
- Streckeneditor: Einstellen von Eigenschaften verschachtelter Gruppenobjekte
- Führerstandseditor: Undo/Redo

## 10.8 Version 2.9

9. November 2014

### 10.8.1 Neue Funktionen

- Gruppenobjekte können selbst Gruppenobjekte enthalten
- Unterstützung externer 3D-Objektmodellformate

- Möglichkeit Objekte auf Strecke an beliebiger Achse zu wiederholen
- Möglichkeit bei Objekten Normalenvektoren pro Fläche und nicht pro Punkt zu definieren
- Funktion "Berechnung Normalenvektoren"
- Funktion "Gruppenobjekt/Objekt am Nullpunkt zentrieren"
- Funktion "Rückseite einer Fläche automatisch erstellen"
- Überarbeitung und Erweiterung der Sichtbarkeitssteuerung
- Warnung beim Überschreiben fremder Dateien
- Neue Funktion *Dateiabhängigkeiten anzeigen* im Expertenmodus des PackageManager verfügbar
- Konvertieren externer 3D-Objektmodellformate ins Loxsim-Format

### 10.8.2 Kleinere Änderungen

- Neue Systemvoraussetzung Windows Vista SP2 oder neuer
- Möglichkeit Streckenmeter als Objekteigenschaft zu benutzen
- Variablen FahrplanDaten::Halt, BedarfshaltBahnsteig und BedarfshaltZug hinzugefügt
- Neuer Operand FahrplanDaten::LastHalt für dynamische Schriften
- Parameter von dynamischen Schriften können im Streckeneditor gesetzt werden
- Objektflächen müssen nicht konvex sein
- Performanceverbesserungen
- Objekte tauchen nicht mehr so plötzlich aus dem Nichts auf
- Fehlende Dateien beim Erstellen eines Package werden angezeigt
- Dialog Beleuchtung: "Slider" auf "SpinBox" geändert
- Wegfall der Option *Graphik unter x fps vereinfachen*
- Neue Expertenoption zur Steuerung der Objektsichtweiten
- Neue Option Windows 8 Vollbildmodus

### 10.8.3 Fehlerkorrekturen

- AFB ohne LZB-Führung nur bis 160 stellbar
- Korrektur falsche Befreiungsmöglichkeit PZB90 1000Hz Beeinflussung

- Zugsicherung Fahrsperr: Überwachungsgeschwindigkeit 2000Hz von 40km/h auf 10km/h verändert
- Probleme Nachtführerstand und Standardgleis behoben
- Korrektur OLE-Variable Wechselblinken
- Fehler im PackageManager bei Deinstallation von Packages behoben
- PZB-Befehl über Joystick wieder möglich
- Editieren von Instrument Weglängenmessung korrigiert
- Fehler beim Abspeichern benutzerdefinierter Charakteristik der Fahrstufen korrigiert

## 10.9 Version 2.8.3

2. April 2014

### 10.9.1 Neue Funktionen

- Transparenz aus Alphakanal
- Zuglängenzähler
- Signal - Option Gruppensignal
- Signalooptionen - gleisabhängiges Zusatzsignal
- Sichtbarkeitssteuerung - Variable VsigKennzahlKleiner
- Nachtführerstandsbild

### 10.9.2 Kleinere Änderungen

- Störschalter für PZB90
- Verbesserung Verhalten PZB90 während LZB-Betrieb
- Wegfall des 16px Rands für transparente Texturen
- Dialog für den Aufruf der Hilfedateien
- Erweiterung des Fahrtenschreibers um PZB-Tasten und LZB-Status
- Anpassung der Uhrzeit-Werte im TCP-Protokoll an die Zusi2-Ausgaben (IDs 10, 11, 12, 50)
- Digitale Instrumente können rechtsbündig dargestellt werden
- Bei fehlender Fahrplandarstellung wird der erfolgreiche Halt im Buchfahrplan/EBuLa angezeigt
- Unverriegelte Türen öffnen bei jedem Halt (auch auf freier Strecke) automatisch

### 10.9.3 Fehlerkorrekturen

- Sound Ende 500Hz Überwachung
- Fehlerkorrektur bei Überlagerung zweier 1000Hz Beeinflussungen (PZB90)
- Korrekturen bei G- und S-Melder im LZB-Betrieb
- Problem bei gleisabhängiger Sichtbarkeitssteuerung behoben

### 10.10 Version 2.8.2a

26. Juli 2013

- Geschwindigkeiten über 160km/h mit LZB wieder möglich
- Bedingtes Abspielen von Streckensounds

### 10.11 Version 2.8.2

18. Juli 2013

- Bei der Darstellung des Buchfahrplans können die Spalten vier und fünf auf Rechtsbündig gesetzt werden.
- Die Darstellung des Buchfahrplans kann auf EBUa umgestellt werden. Die EBUa-Anzeige ist nur vorbildähnlich.
- 2D-Fonts können sämtliche Zeichen aus Unicode enthalten
- Gefahrene Km bzw. Anzahl Aufrufe werden auch pro Fahrplan gespeichert
- Die Anzahl der Zeilen des Buchfahrplans im Führerstand kann nun im Editor gesetzt werden.
- Komplette Neuimplementierung der PZB90
- Stadtbahn PZB90
- Englische Version des Simulators
- PackageManager: Deinstallation von Packages während Installation neuer Packages möglich
- Joystick Achsen umkehrbar
- Schneefall (Alpha Status) über Wetterdateien steuerbar
- Der Indusimagnet des Schutzsignals wird eigenständig ausgewertet
- Lüfter läuft nur bei Verwendung dynamischer Bremse nach
- Lüfter schaltet mit HS aus
- Anzeige Lüfterstatus verzögert anhand von Sound
- Testen im Loks>Edit für Wetterdateien, Dynamische Schriften u. Dynamische Sichtbarkeitssteuerung verbessert
- Kameraposition im (Gruppen)objekt-Editor bleibt nach Update-Button erhalten
- Leerzeilen in der Buchfahrplananzeige unterdrückt

- Gruppenobjekt-Editor: Eigenschaften von Gruppenobjekt "überleben" ein Refresh der 3D-View
- PackageManager: Anzeige von überschriebenen oder gelöschten schreibgeschützten Dateien ganz oben
- Zufall pro Gruppenobjekt mit Sonstige::ZufallGruppenObjekt
- CrashReport Sprache wird dynamisch anhand ausgewählter Sprache bei Installation bzw. PackageManager gesetzt
- Sondersounds werden mit der Lautstärke für Ansagen abgespielt
- Standard Sky-Datei wird nicht mehr automatisch im Fpl gesetzt
- Darstellung Bedarfshalt in der Fpl-Anzeige geändert.
- LoksimEdit: Datei-Öffnen Dialog enthält "Doku" Button
- Verbesserte Auswahl für Standardauflösung und 3D-Treiber
- Fette Überschriften im Lokeditor
- Multimonitor-Support bei gleichen Treibernamen
- TCP: Soll-Fahrstufe und Oberstrom
- Bugfix Öffnen SZuletzt verwendeter Datei "die nicht mehr existiert"
- PackageManager - Installation rückgängig machen bei schreibgeschützter Datei
- Fehler bei Speichern unter - Textur kopieren behoben
- Tippfehler und Texte im Lokeditor überarbeitet
- 'Verschleppte' Bezeichnungen aus dem Nebengleis gefixed
- Leerzeilen im Buchfahrplan Führerstand unterdrückt
- Fehler in L3dEditLauncher behoben (mehrere Loksim Installationen)
- Fehler in der Fahrplananzeige behoben
- Mauseinstellung
- Verzerrter Sound im Stand
- Korrektur Anzeige Bedarfshalt im Fst (manchmal zu früh)
- Joystick mit Funktion "Kombibremshel (inkl. Beschleunigung)" bei Fst. mit Kombibremshel verwendet
- Diverse Korrekturen bei LZB und AFB
- Doppelter Nullstellzwang
- Zs1 im LZB-Betrieb
- AFB + Kombihebeler

#### **Version 2.8.1a** 07. Dezember 2012

- Variable Sonstige::Zuglaenge
- Senden der Daten in Fehlerberichten optimiert
- Option Texturhandling standardmäßig auf "Bei Bedarf laden und nicht im Speicher halten"
- AFB nur bis max. 160km/h ohne LZB

- Bugfix Haltansagen
- Bitmap bei Font Erstellen wird wieder automatisch generiert
- PackageManager funktioniert auch auf FAT32 Partitionen
- Kleinere Bugfixes im PackageManager

#### **Version 2.8.1** 26. Oktober 2012

- Die erste und zweite Spalte in der Fahrplandarstellungen der Führerstände sind jetzt als rechtsbündige Ausgabe möglich
- Fonts sind jetzt wie Objekte über statische Zustandsvariablen schaltbar
- Indusi zusätzlich: Nunmehr sind Geschwindigkeitsprüfabschnitte auch signalabhängig möglich
- In Fahrplänen können benutzerdefinierte Variablen zur Verwendung in logischen Ausdrücken gesetzt werden
- Trennung von Daten- und Programmverzeichnis möglich
- Automatisches Erstellen von Fehlerberichten
- Ein durchfahrener Halt kann als Zugfolgestelle definiert werden.
- Ein Halt kann als Betriebshalt definiert werden.
- Kachelung von Texturen in Skyboxen möglich
- Texturen im PNG- und TGA-Format werden unterstützt
- Dynamische Schriften möglich
- LZB Verbesserungen: BKW und realistischere Bremskurven
- Leuchtmelder Halbstufe reaktiviert
- Steuerung S-Melder angepasst
- Berechnung der Zuglänge im Bremszettel geändert
- (Dis)connect Buttons für TCP-Anbindung eingebaut
- PackageManager an kleinere Auflösungen angepasst
- PackageManager kann installierte Dateien mit 'Doppelklick' sofort öffnen
- Quadratische Texturen bei seitlichen Flächen in Skyboxen möglich und empfohlen
- Bei Wetterauswahldialog ist standardmäßig SZufällig"(Skybox) ausgewählt und nicht mehr die klassische Steuerung
- Loksim(Edit) About-Dialog zeigt Verwendung von SSE2 an
- Gebaeude1\_FFS von RainerH in Standard-Package 2.8.1 inkludiert
- Anpassung an Benutzerkontensteuerung
- Beim Package Erstellen kann man nun standardmäßig auch .txt, .pdf, .xps Dateien auswählen (Doku)
- Über den Eigenschaften-Dialog kann man im LokimEdit jeder Datei eine Doku zuweisen. Für Fahrpläne und Loks wird im Loksim3D ein "Doku"Button zum Öffnen dieser Doku angezeigt
- Joystick Slider (Schubregler) ist nutzbar



- Bedarfshaltanzeiger kann im Editor mittels SSignal grün/rot"Button getestet werden
- Bedarfshalt immer für Testzwecke einstellbar
- Zeitpunkt an dem Bedarfshalt-Anzeige im Fst aufleuchtet wird zufällig bestimmt
- Installer registriert automatisch LoksimControl.exe
- Speicherlimit für Loksim3D bzw. LoksimEdit auf 3 bzw. 4 GB angehoben (32 bzw. 64 Bit OS)
- Erweiterung der logischen Ausdrücke
  - FahrplanDaten::Ankunftszeit"<halt>"
  - FahrplanDaten::Abfahrtszeit"<halt>"
  - FahrplanVars::
  - Funktionen::TimeDif(<arg1>, <arg2>)
- Neue Mausgesten bei der 3D Vorschau von Objekten
- Normalenvektoren können im (Gruppen-)Objekteditor ausgeblendet werden
- PreviewHandler zeigt Readme von .l3dpack Dateien
- Joystickfunktionen Zugkraftregler (+/-) und AFB-Ziel (+/-) an Verhalten von Fahrstufe (+/-) angeglichen
- Neue Icons
- Vorsichtsignal (Zs7) ermöglicht ebenfalls die Abfahrt
- Anzeige Indusi-Art in Fahrtenschreiber + LokInfo Anzeige korrigiert
- Installer räumt Registry- und Startmenüeinträge bzw Dateien des alten Installers (2.7.2) auf
- Abstürze des WetterEditors behoben
- Streckensound im selben Ordner wie .l3dstr-Datei nun möglich
- Ende Fahrt Anzeige überdeckt nicht mehr Fahrtauswertung
- Diverse Tippfehler korrigiert
- Bugfix für Anzeige Schnellbremsung bei OLE bzw. TCP
- keine Soundkarte Fehlermeldung wird nur 1x gezeigt
- Absturz LoksimEdit bei Verwendung von Touch-Monitor behoben
- Fehler bei F11 bei kurzen Haltabständen behoben
- Fehler bei Planabfahrt kurz nach 00:00 behoben

## Version 2.8 11. März 2012

- Zusätzliche und erweiterte Signalfunktionen
- Geschwindigkeitsprüfabchnitte
- Logische Ausdrücke zur Sichtbarkeitssteuerung bzw Soundsteuerung
- Wetter- / Himmelstextursteuerung
- In Fahrplänen kann ein Sound definiert werden, welcher in

einer bestimmten Entfernung vor einem (Bedarfs)halt abgespielt wird

- Standarddateidialog auch unter Windows XP verwendbar
- Neuer PackageManager für die (De)installation von Loksim-Packages
- BÜ-Namen können nicht nur aus einer Liste ausgewählt werden, sondern auch direkt per Namen eingegeben werden (sinnvoll falls BÜ in anderer Streckendatei definiert ist)
- Vertikaler Schriftzug bei Gruppenobjekten
- Neue Hauptsignaleigenschaft als Checkbox: Zwischensignal
- Einbau CH-Sifa mit und ohne HS-Auslösung. Minor Bugfixes CH-Indusi und neu Auslösung mit oder ohne HS-Auslösung.
- Fehlende Verzögerung zwischen Schalten von BueLicht und BueSchranke bei Bahnübergängen eingebaut
- [Vista+] Dateivorschau Handler (PreviewHandler) wird bei Installation registriert
- [Vista+] Aufnahme der Loksim-Dateien in Windows-Suchindex bei Installation
- Programmende beim Überfahren eines Rot zeigenden Signales ist abschaltbar
- Erweiterung des Kombihbels um die Option nur dyn. Bremse".
- Fehlerkorrektur: Keine vZielüberwachung in der PZB bei Wechselblinken ohne 500er und 1000er-Melder.
- Fehlerkorrektur: Nullstellungszwangauflösung über Zugkraft 0 möglich.
- Joysticksteuerung bei Beschl+Bremsen mit Zugkraftregler korrigiert
- Endlosschleifen KBS-Editor Weichenstellung verhindert
- Bei der Zifferneingabe im Editor werden jetzt auch Kommas (Beistriche) akzeptiert
- Real-Sifataste auch auf den Achsen der Joysticksteuerung
- Bugfix beim Bestimmen des relativen Pfads von Loksim-Dateien gegenüber anderen Loksim-Dateien
- Die Ausgabe 'Streckenlimit' so gesetzt, das auch ein Limit von 99km/h die Kennziffer 9 ergibt.
- Standard-3D Einstellungen geändert: Max-Texturgröße 1024, Alle Texturen Laden und im Speicher halten, Hohe Farbtiefe, Cache nicht verwenden
- L3dEditLauncher startet bei Vorhandensein mehrerer Loksim-Installationen immer den Editor im richtigen Ordner
- Unterstützung von CrashDumps
- Einsatz des SSE2 Befehlssatz bei neueren CPUs

#### **Version 2.7.2** 12. Dezember 2010

- Bug Lok Editor behoben (fehlende Eingabefelder für Instru-

mente)

#### **Version 2.7.1** 29. November 2010

- Erweiterte Bahnübergangssteuerung
- Es kann der Sifa/Indusi-Zwangsbremzungssound mit der Option nur einmal"versehen werden.
- Auswertungs Bitmap funktioniert nun auch bei GDI Darstellung
- Bug bei Laufleistung km Protokollierung behoben (alte Statistiken werden automatisch gelöscht)
- Button Texturecache löschen funktioniert wieder
- Streckensounds bei mehreren Modulen funktioniert nun
- Bug in der Fahrtenschreiberwertung behoben (- Prozente)
- Bug bei Störungshäufigkeit von Signalgedeckten und Signalgedeckt (Streckenblock) BÜs
- Standard Dateinfo und -autor kann angegeben werden
- Ab Vista kann anstatt dem loksimspezifischem der Standard Windows Dialog verwendet werden
- Anzeige von Limits bei denen Limit nicht anzeigen"gesetzt ist, werden standardmäßig in der EBULA nicht mehr angezeigt. Für Anfänger gibt es die Möglichkeit diese realitätssteigernde Option auszuschalten
- Infofeld bei Fahrplan- und Lokauswahl vergrößert
- Bug Haltestellenansage nach letztem Halt behoben
- Bug Zs9 Meldung kommt nur mehr bei gestörten BÜs
- PZB Befehl auf Maustaste möglich
- Unnötige Änderungen übernehmen Nachfrage im Editor bei Weitsichtbar behoben
- Streckensounds im Editor standardmäßig stummgeschaltet; Zustand der SStummschaltung" wird bei Ein- und Ausschalten der Vorschau übernommen
- Einbau SBB-Signum
- Bug Uninstall behoben
- Farbe und Breite für Sekundenzeiger in Analoguhr einstellbar
- Bei Package-Installation werden die Zeitstempel bei versuchtem Überschreiben von schreibgeschützten Dateien angezeigt
- Bug bei Zs1 behoben

#### **Version 2.7** 1. Juli 2010

- 2D Darstellung standardmäßig mit DirectX
- Möglichkeit Sounds auf Strecke einzubinden
- Fehlerkorrekturen bei Packageinstallation