

# Loksim3D Funktionsdokumentation

Ralf Gryga

Lukas Haselsteiner  
Martin Mory

Uwe Klein

März 2014

---

# 1 Vorwort

---

Herzlich Willkommen zur Funktionsdokumentation des Loksim3D. Diese Dokumentation stellt im Moment eine Beschreibung neuer Funktionen ab Version 2.8.2 dar. Im Laufe der Zeit soll diese Dokumentation auch um bestehende Funktionen erweitert werden, und so ein umfassendes Nachschlagewerk für sämtliche Funktionen des Loksim3D werden. Sie sollte damit in Zukunft *die* Beschreibung für erfahrende Loksimmer darstellen und soll insbesondere möglichst viel Informationen zum Objekt-, Führerstand- und Streckenbau bieten.

Wir hoffen, dass nach dem Studium der Einstiegsdokumentation 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: Dafür ist Erfahrung im Programmieren erforderlich. Insbesondere 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
- ...

## 1.1 Versionshistorie

### 1.1.1 Version 2.8.3

#### Neue Funktionen

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

#### 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

#### Fehlerkorrekturen

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

### 1.1.2 Version 2.8.2a

**26. Juli 2013**

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

- Bei der Darstellung des Buchfahrplans können die Spalten vier und fünf auf Rechtsbündig gesetzt werden.
- Die Darstellung des Buchfahrplans kann auf EBUla umgestellt werden. Die EBUla-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
- Englische Version des Simulators
- PackageManager: Deinstallation von Packages während Installation neuer Packages möglich
- Stadtbahn PZB90
- 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" 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.
- Loks>Edit: 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 exi-

stiert

- 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
- Mausradsteuerung
- Verzerrter Sound im Stand
- Korrektur Anzeige Bedarfshalt im Fst (manchmal zu früh)
- Joystick mit Funktion "Kombibremshebel (inkl. Beschleunigung)" bei Fst. mit Kombibremshelben verwendbar
- Diverse Korrekturen bei LZB und AFB
- Doppelter Nullstellzwang
- Zs1 im LZB-Betrieb
- AFB + Kombihebel

#### **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 Geschwindigkeitsprüfabchnitte 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 "Signal 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 LoksिमEdit 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
- Geschwindigkeitsprüfabschnitte
- 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
- Standarddateialog auch unter Windows XP verwendbar
- Neuer PackageManager für die (De)installation von LoksिमPackages
- 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 Loksिम-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 Instrumente)

**Version 2.7.1** 29. November 2010

- Erweiterte Bahnübergangssteuerung
- Es kann der Sifa/Indusi-Zwangsbremssound 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 Fahrtenschreiberauswertung 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

---

# Inhaltsverzeichnis

---

<b>1</b>	<b>Vorwort</b>	<b>1</b>
1.1	Versionshistorie . . . . .	2
1.1.1	Version 2.8.3 . . . . .	2
1.1.2	Version 2.8.2a            26. Juli 2013 . . . . .	2
1.1.3	Version 2.8.2            18. Juli 2013 . . . . .	3
<b>I</b>	<b>Simulator</b>	<b>12</b>
<b>2</b>	<b>Steuerung</b>	<b>13</b>
2.1	Diverses . . . . .	13
2.1.1	Zuglängenzähler / Wegmessung . . . . .	13
<b>3</b>	<b>Optionen</b>	<b>14</b>
3.1	Simulation . . . . .	14
<b>4</b>	<b>PackageManager</b>	<b>15</b>
4.1	Übersicht . . . . .	15
4.2	Funktionsweise . . . . .	15
4.3	Deinstallation während Installation . . . . .	16
<b>II</b>	<b>Editor</b>	<b>17</b>
<b>5</b>	<b>Objekteigenschaften</b>	<b>18</b>
5.1	Logische Ausdrücke . . . . .	18
5.1.1	Operanden (Variablen) . . . . .	18
5.1.2	Operatoren . . . . .	19
5.1.3	Funktionen . . . . .	20
5.1.4	Beispiele . . . . .	20
5.1.5	Einfluss auf Performance . . . . .	21

5.2	Dynamische Schriften . . . . .	22
5.2.1	Syntax in EBNF . . . . .	22
5.2.2	Erklärung . . . . .	23
5.2.3	Performance . . . . .	24
5.2.4	Beispiele . . . . .	24
5.3	Sichtbarkeitssteuerung . . . . .	25
5.3.1	nur sichtbar bei / nur unsichtbar bei . . . . .	25
<b>6</b>	<b>Strecke</b>	<b>26</b>
6.1	Gleis . . . . .	26
6.1.1	Gleiseigenschaften . . . . .	26
<b>7</b>	<b>Lokeditor</b>	<b>28</b>
7.1	Führerstand . . . . .	28
7.2	Dialog Motor (Antriebsdaten) . . . . .	28
7.2.1	maximale Geschwindigkeit (km/h) . . . . .	29
7.2.2	maximale Motorspannung (V) . . . . .	29
7.2.3	Wirkungsgrad der Motoren (%) . . . . .	29
7.2.4	maximale Lokleistung (kW) . . . . .	29
7.2.5	Maximaler Motorstrom (A) . . . . .	30
7.2.6	Wirkungsgrad Transformator (%) . . . . .	30
7.2.7	Oberspannung (V) . . . . .	30
7.2.8	maximale Zugkraft (kN) . . . . .	30
7.2.9	Motorparameter . . . . .	30
7.2.10	Standardwerte für E-Loks, Drehstromloks und Dieselloks . . . . .	32
7.2.11	Diagramm . . . . .	33
7.2.12	Bemerkung . . . . .	33
7.3	Indusi/PZB . . . . .	33
7.3.1	Art der Indusi . . . . .	33
7.3.2	Sounds . . . . .	34
7.3.3	Leuchtmelder und Anzeigen . . . . .	34
7.3.4	PZB Einstellungen . . . . .	34
7.4	Türsteuerung . . . . .	36

7.5	Diverses . . . . .	36
7.5.1	Wegmessung / Zuglängenzähler . . . . .	36
<b>8</b>	<b>Objektor</b>	<b>37</b>
8.1	Textur . . . . .	37
8.1.1	Transparenz . . . . .	37
8.1.2	Empfehlungen . . . . .	38
<b>9</b>	<b>Wettereditor</b>	<b>39</b>
9.1	Niederschlag . . . . .	39
<b>10</b>	<b>Allgemeines</b>	<b>40</b>
10.1	Package erzeugen . . . . .	40
10.1.1	Diese Dateien löschen / Packages deinstallieren	40

---

## 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 Tfz das Ende des Gefahrenbereichs erreicht, startet der Tfzf die Wegmessung. Nachdem die Lok eine Zuglänge zurückgelegt hat, ertönt ein akkustisches Signal. Dadurch weiß der Tfzf, 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 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

---

## 4 PackageManager

---

### 4.1 Übersicht

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

Im Normalfall wird beim Öffnen einer `.l3dpack` Datei automatisch der PackageManager gestartet. Ein Klick auf Installation starten 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 Loxsim-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 Loxsim-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 Loxsim Verzeichnis zurück. Bei Führerständen und Fahrplänen kann dies sogar zur Verwirrung des Benutzers führen, bei sämtlichen anderen Dateien bleiben unschöne "Leichen" im Loxsim 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 Objekteigenschaften

---

### 5.1 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.

#### 5.1.1 Operanden (Variablen)

##### Konstante

Es können konstante, *ganzzahlige* Zahlen als Operatoren verwendet werden. Zusätzlich gibt es die Möglichkeit Zeitpunkte als hh:mm:ss oder hh::mm zu definieren (Stunden:Minuten:Sekunden). Diese Schreibweise wird bei der Auswertung in Sekunden nach 00:00 Uhr umgerechnet.

##### 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:

**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

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

**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 StrgQ *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

### 5.1.2 Operatoren

Operatoren verknüpfen die Operanden miteinander um das Endergebnis zu berechnen. Auflistung der verfügbaren Operatoren nach *absteigender* Priorität angeordnet (gleiche Hintergrundfarbe bedeutet gleiche Priorität):

Operator	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

### 5.1.3 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(int EndeSek, int BeginnSek)
{
    Falls (EndeSek < BeginnSek)
    {
        EndeSek = EndeSek + SEKUNDEN_PRO_TAG
    }
    Ergebnis = EndeSek - BeginnSek
    return Ergebnis
}
```

### 5.1.4 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)
```

### 5.1.5 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!

## 5.2 Dynamische Schriften

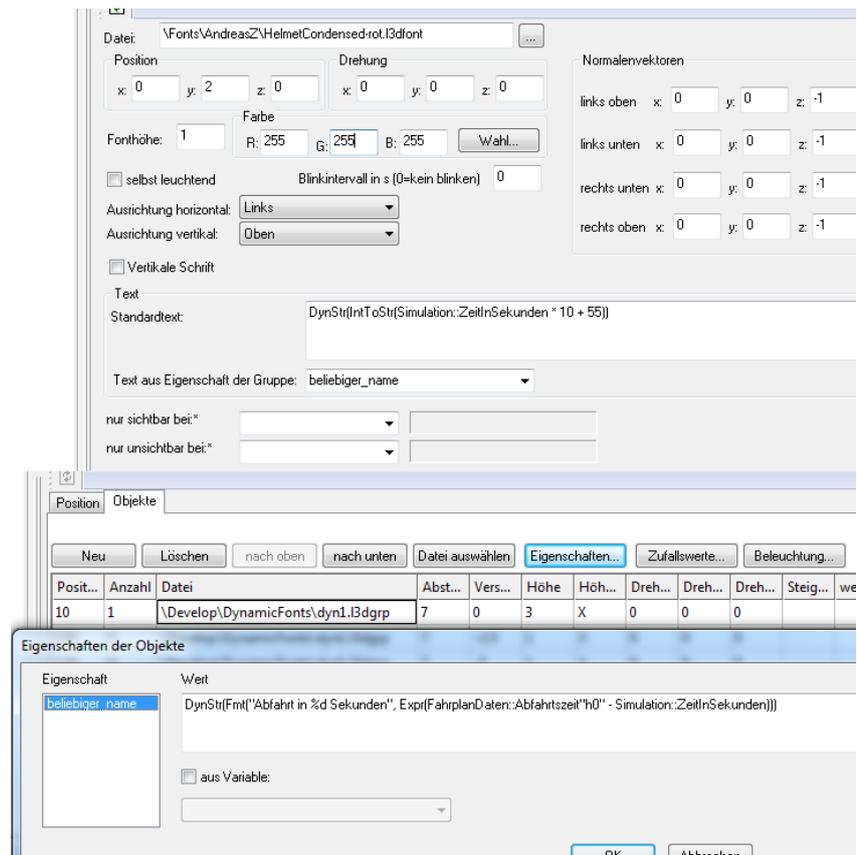


Abbildung 5.1: Möglichkeiten dynamische Schriften zu setzen (oben Gruppenobjekt, unten Streckeneditor)

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.

### 5.2.1 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, ')'</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 Variablenamen
<code>fmt_chars</code>	=	Gültiger printf Format-String
<code>halt_chars</code>	=	Gültiger Haltname, Etwaige Anführungszeichen im Haltnamen müssen durch <code>\</code> ersetzt werden

---

Tabelle 5.2: Syntax der Dynamischen Schriften in EBNF

## 5.2.2 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.

---

Ausdruck	Ergebnis
----------	----------

---



## 5.3 Sichtbarkeitssteuerung

### 5.3.1 nur sichtbar bei / nur unsichtbar bei

**VorSigKennzahlKleiner** 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 VorSigKennzahlKleiner FALSE.

---

## 6 Strecke

---

### 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). Häkchen 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 (Durchfahr Gleis) 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 "GIAbhZSigäuswählen".
- In den Eigenschaften einer Strecke an der gewünschten Position eine neue Signaloption definieren und bei "Gleisabhängiges Zusatzsignal" das Häkchen setzen. Beim Eintrag Referenzzeichen ein Zeichen (Buchstabe, Zahl) eintragen, das dem gewünschten Fontzeichen entspricht.

---

## 7 Lokeditor

---

### 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 Loks-Funktion *Motorstrom berechnen* benutzt, berechnet Loks 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

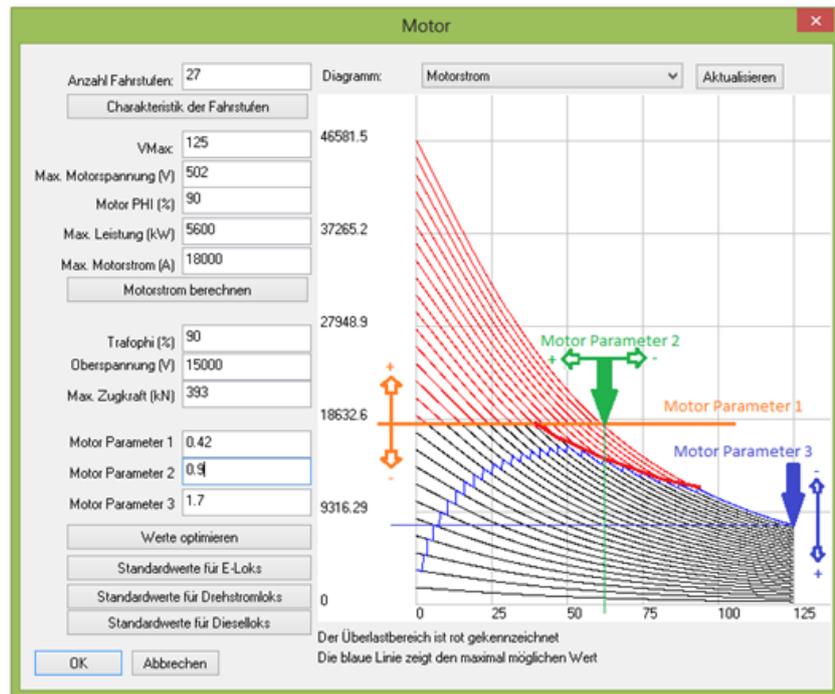


Abbildung 7.1: Motorparameter der Ae 610 der SBB

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 Loksिम 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 Loksिम 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. Loksिम 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.

Störschalter, 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 Objekteditor

---

### 8.1 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.

#### 8.1.1 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.

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

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.

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

### 8.1.2 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 2096px), 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

---

## 9 Wettereditor

---

### 9.1 Niederschlag

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

*Die Umsetzung des Niederschlags im Loxim 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

---

## 10 Allgemeines

---

### 10.1 Package erzeugen

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

#### 10.1.1 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