

# STL koppel och slutsignaler PBR

För STL-koppel v4.64.0



Denna manual behandlar de nya meshbiblioteken med koppel och slutsignaler gjorda med moderna tekniker som PBR-texturer och LOD till STL-koppel-systemet. Dessa nya meshbibliotek **fungerar enbart i TRS2019 och uppåt**, men de gamla meshbiblioteken finns kvar så att gamla vagnar fortsätter att se ut och fungera som de gjort tidigare.

## Bakgrund

STL-koppel har sedan tidigare bestått av tre delar, nämligen:

- **STL Script Library <kuid2:177292:209000:X>** Scriptbiblioteket som implementerar själva STL-koppel funktionaliteten.
- **STL Koppel LOD 4.3 (<kuid2:67375:1021:X>)** som innehåller normalspårskoppel och slutsignaler och -hållare av olika slag och årgångar.
- **STL NGKoppel LOD 4.3 (Narrow Gauge, <kuid2:67375:100437:X>)** med extra modeller på smalspårskoppel och kopplingsslangar till tryckluftsbröms för smalspåriga fordon, framförallt för 891mm spårvidd.

De två sista av dessa, meshbiblioteken, innehåller modeller på slutsignaler ursprungligen gjorda av STW (Svenska Tåg Werkstäderna), samt koppel och andra kompletterande modeller av Lan och Korvtiger från STL. Själva scriptet som sköter animationer av koppel, övergångsbryggor och annat, samt att sätta ut rätt slutsignaler är till största delen skrivet av Svenolov med mindre modifikationer av Korvtiger.

Sedan ett par versioner tillbaka av Trainz så tvingas användandet av LOD:ade mesher (via .lm.txt-filer) om man vill använda de senaste funktionerna i sina assets till Trainz. LOD står för Level Of Detail och innebär att varje mesh görs i flera varianter med olika detaljgrad där mer enkla (högre) LOD-nivåer väljs ju längre bort meshen visas i spelet. Det gjorde att vi i STL med hjälp av Vovven47 tillförde "fusk-LOD:ade" varianter av alla våra mesher i dessa koppelbibliotek, dvs. .lm-filer där den detaljerade gamla meshen bara byttes mot en kloss när meshen var så långt bort att den inte syntes längre, för att man skulle kunna använda dessa koppel och slutsignalmesher i fordon som var byggda med moderna tekniker. Detta är en "lösning" som fungerar, men som inte drar nytta av de prestandaförhöjande effekter som LOD är menat att ge, då alla mesher var fullt detaljerade nästan hela tiden.

Med releasen av Trainz Railroad Simulator 2019 (TRS2019) så lade man till stöd för PBR-texturer, ett helt nytt sätt att texturera på som ger betydligt högre realism till material än tidigare textureringsmodell som alla gamla STL-koppelmesher är gjorda med.

Kombinationen av dessa två begränsningar av de gamla meshbiblioteken i STL-koppel, tillsammans med det faktum att vi idag förväntar oss ännu mer detaljerade modeller än då dessa bibliotek skapades, har lett fram till två nya meshbibliotek för koppel och slutsignaler, där alla mesher är byggda med både korrekt LOD:ning och med PBR-texturer, samt har en högre detaljeringsgrad än tidigare bibliotek. Dessa nya meshbibliotek är skapade av Korvtiger och Tanigardi. Meshbiblioteken använder samma STL-koppel-script som de gamla modellerna, så det handlar alltså bara om nya mesher, även om själva STL-koppelscriptet fått några mindre uppdateringar. De gamla mesh-biblioteken kommer att fortsätta att finnas kvar och fungera som förut med de gamla fordon som använder det. Det rekommenderas dock att alla nybyggda modeller (de som är byggda för TRS2019 och senare versioner av Trainz) använder de nya meshbiblioteken. Gamla modeller kan oftast konverteras på ett

enkelt sätt till att använda de nya meshbiblioteken genom att bara göra några mindre ändringar i config-filen på assetsen.

Precis som tidigare kan modellbyggare som önskar göra egna modeller av slutsignaler eller koppel bygga sådana och använda istället eller tillsammans med mesherna i dessa bibliotek.

Lite information om det finns senare i detta dokument.

Namn	STL Koppel LOD 4.3	STL NGKoppel LOD 4.3	Svenska Koppel och slutsignaler PBR	Svenska Smalspårskoppel PBR.
Kuid	<kuid2:67375:1021:X>	<kuid2:67375:100437:X>	<kuid2:328014:599:X>	<kuid2:267353:101057:X>
LOD	Nja, Fusk-LOD	Nja, Fusk-LOD	Ja	Ja
PBR-texturer	Nej	Nej	Ja	Ja
Animerade mesher	Ja	Ja	Ja	Ja
Fungerar i versioner till och med TANE	Ja	Ja	Nej	Nej
Fungerar i TRS2019 och senare	Ja	Ja	Ja	Ja

## Innehåll normalspårskoppel och slutsignaler

De nya biblioteket för normalspårskoppel och slutsignaler heter **Svenska Koppel och slutsignaler PBR <kuid2:328014:599:X>** och innehåller följande mesher:

### Normalspårskoppel till tvåaxliga fordon

Meshnamn: `koppel_2l.lm`

Normalspårskoppel av 1898 års modell med fästplatta mot buffertbalken av den fasta varianten som användes på lok, samt kortare tvåaxliga vagnar och där fordonet inte var så pass långt att dragstången behövde ledas. Senare togs andra modeller av koppel fram med något grövre dimensioner.



### Normalspårskoppel till boggievagnar

Meshnamn: `koppel_bo.lm`

Normalspårskoppel av 1898 års modell med ledad dragstång och avlång öppning i buffertbalken. Denna typ användes på boggievagnar och en del andra längre tvåaxliga fordon som var så pass långa att de behövde en dragkrok som kunde röra sig lite i sidled.



### Slangkopplingar till tryckluftsbroms för normalspåriga lok

Meshnamn: `luft_0_lok.lm` och `luft_1_lok.lm`

Slangkoppling för bromsledningen till tryckluftsbroms med blikkopplingar av den typ som användes på lok. Dessa blindkopplingar täppte till öppningen på kopplingsnåven för att skydda den från smuts och skador. Lok hade i regel alltid två slangar i vardera ände, för att alltid ha en i reserv, ifall en näve skulle bli obrukbar.



## Slangkopplingar till tryckluftsbroms för normalspåriga vagnar

Meshnamn: `luft_0_vagn.lm` och `luft_1_vagn.lm`

Slangkoppling för bromsledningen till tryckluftsbroms med den enklare typ av upphängningsanordning som användes på både godsvagnar och personvagnar. Godsvagnar hade i regel enbart en slang i vardera ände, medan personvagnar i regel hade två per ände. Detta dels för att ha en i reserv, men också för att underlätta vid ihopkoppling genom att kunna koppla på samma sida då övergångsbryggor, bälgar och ångvärmeledningar annars ofta satt ivägen. Även godsvagnar som ofta kopplades in i persontåg fick dubbla kopplingsslangar.



## Slutsignalhållare

Meshnamn: `ssh_v.lm` och `ssh_h.lm`

Slutsignalhållarna är precis som namnet antyder en hållare för slutsignalerna, alltså de skärmar eller lyktor som sattes upp på den sista vagnen i tåget för att märka ut slutet på tåget. Denna modell av hållare användes på nylevererade vagnar åtminstone från slutet på 1800-talet fram till 1950-talet. Hållarna monterades på långsidorna, vid gavlarna. Tvåaxliga vagnar fick i regel bara hållare i vagnens ena ände, medan boggievagnarna fick det i båda ändarna för att man skulle kunna sätta dem i den bakre änden av vagnen. En del vagnar fick speciallösningar med lösa slutsignalhållare, eller hållare som kunde fällas in när de inte användes, för att inte vara i vägen.



Samma modell användes både på normalspårs- och smalspårsfordon.

## Slutsignalskärmar

Meshnamn: `skarm_gul.lm` och `skarm_vit.lm`

Slutsignalskärmar av den typ som användes från 1907. Skärmarna var ursprungligen målade vita med en röd rand, men man övergick under 40-talet till att måla dem röd/gult för att de skulle synas bättre, liksom många andra signalsäkerhetsföremål vid denna tiden. Skärmar användes på tåg som enbart framfördes under dagtid, vid god sikt, annars hade man lyktor. Skärmarna skulle



sitta en på var sida av vagnen för att markera sista vagnen i tåget.

Samma mesh används på båda sidorna av vagnen. Samma modell användes både på normalspårs- och smalspårsfordon.

## Slutsignallykta av 1930-talsmodell

Meshnamn: lykta\_1930t\_v.lm och  
lykta\_1930t\_h.lm

Slutsignallykta med batteri av den modell som infördes runt 30-talet. Lyktorna användes som slutsignaler nattetid och hade både vitt sken framåt i tågets färdriktning och rött bakåt.

Meshen finns i ett höger- och ett västerutförande, för att rätt färg på skenet ska hamna framåt. Samma modell användes både på normalspårs- och smalspårsfordon.



## Innehåll smalspårskoppel

Det nya biblioteket med smalspårskoppel heter **Svenska Smalspårskoppel PBR**, <kuid2:267353:101057:X> och innehåller följande modeller på koppel:

### Hästskokoppel med öppningen uppåt

Meshnamn: `koppel_up.lm`

Kopplet består av en hästhovformad kammare som genomgås av en sprint. I sprinten hänger en koppellänk formad som en åtta som vid ihopkoppling fästes i det andra kopplets sprint. Under kopplet sitter en centrerad säkerhetskätting som är tänkt att hålla ihop fordonen ifall sprintarna eller koppellänken skulle brista.

Modellen användes bland annat på BLJ och DONJ.



### Hästskokoppel med öppningen nedåt

Meshnamn: `koppel_down.lm`

Denna typen fungerar på samma sätt som den uppåtvända, med den enda skillnaden att öppningen är vänd nedåt, vilket kunde vara något knöligare vid ihopkoppling.

Användes bland annat på SRJ och 1067mm-banorna i Småland och Blekinge



### Tvåkammarkoppel med säkerhetskedjor

Meshnamn: `koppel_2chain.lm`

Tvåkammarkoppel, (eller Klemmingkoppel som det också heter efter dess uppfinnare Viktor Klemming) är en vidareutveckling av äldre koppelmodeller där man tog bort de negativa sidorna med de äldre modellerna, (som att enkammarkoppel inte kunde användas för felvända vagnar) samt en enklare koppelmekanism som dessutom var halvautomatisk till skillnad från hästskokopplens koppellänkar. Om hakarna är nedfälda när två fordon körs mot varandra så hakar de automatiskt i



varandra. För att hakarna inte ska riskera att hoppa upp läggs en tyngd i en kätting över haken när den är kopplad. Koppeltypen blev mycket lyckad och användes på många banor. Hos SJ blev den mer eller mindre standard på de övertagna smalspårslinjerna efter förstatligandet. Varianter med en skruv som kunde skruva ut stötplanet testades och användes från och med 1943.

Denna modell har säkerhetskedjor som extra säkerhet ifall båda hakarna skulle brista eller råka haka ur sig.

Denna koppelmanus användes bland annat på Lelångsbanan, FJ, Sävsjö-Vetlanda-Målilla Järnväg och MÅJ.

## Tvåkammarkoppel med skruvkoppel

Meshnamn: `koppel_2screw.lm`

En variant på tvåkammarkopplet ovan, men där säkerhetskopplet består av två skruvkoppel, liknande de på normalspåriga fordon, istället för kedjor.

Denna modell användes på bland annat VGJ, KBJ, VÅHJ, MÖJ och NÖJ.



## Scharfenbergkoppel

Meshnamn: `koppel_sbg.lm`

Scharfenbergkopplet är det kopplet som skiljer sig mest från de övriga och användes sällan på loktåg, utan oftast som motorvagnskoppel, både på smalspår och på normalspår. Koppeltypen används än idag, om än i modifierad form.

Scharfenbergkoppel på olika motorvagnstyper är ofta av skilda modeller, vilket gör att man inte kan koppla ihop två fordon av olika typer hur som helst. På banor där Scharfenberg och andra tågekoppel användes kunde kombinationskoppel, en slags adapter mellan Scharfenbergkopplet och ett vanligt koppel för banan finnas, för att kunna koppla ihop en motorvagn med en vanlig person- eller godsvagn som släpvagn eller för transport vid maskinskada.



Denna modell av Scharfenbergkoppel är den som satt på de smalspåriga Hilding Carlsson-rälsbussarna

## Slangkopplingar till tryckluftsbröms för smalspåriga fordon

Meshnamn: luft\_0\_anim.lm och luft\_1\_anim.lm

Även smalspåriga banor fick genomgående bromssystem av samma typ som de normalspåriga, dock så infördes det senare än på normalspåret, då de smalspåriga banorna var privata och hade sämre ekonomi än SJ. Precis som på normalspåret så hade lok, personvagnar samt vagnar som ofta gick i personförande tåg två kopplingsslangar i varje ände och övriga vagnar (godsvagnar) endast en. Detta för att ha i reserv på personvagnar som var svårare att byta ut eller koppla ur ett tåg och för att bryggorna, ångvärmeledningar och annat kunde vara i vägen mellan vagnarna.

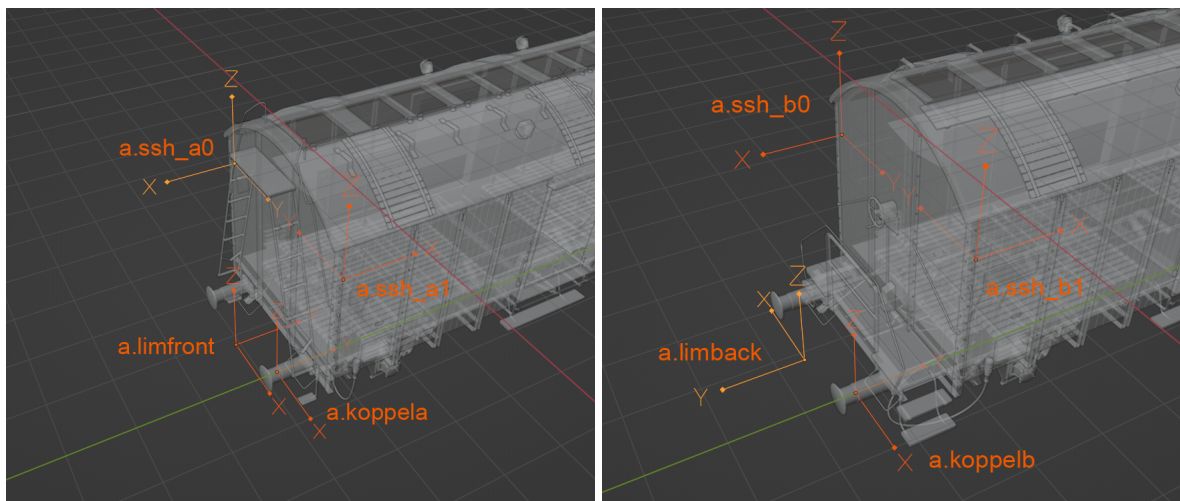


Modellen på enkammarkoppel som finns i det äldre mesh-paketet har utelämnats från detta nyare paket då få banor använde den typen av koppel under en längre tid.

## Attachments

För att kunna montera de olika delarna på fordonet behövs några attachments. Alla behövs inte på alla fordon, det beror på vilka val man gör av slutsignaler och annat. Följande attachments finns:

- **a.koppela** - fästpunkt för koppel, tryckluftsbromsslangar, övergångsbryggor, bälgar och annat i A-änden (a.limfront-änden)
- **a.koppelb** - dito i B-änden (a.limback-änden)
- **a.ssh\_a0** - fästpunkt för slutsignalhållare och slutsignaler i A-änden (a.limfront) på vänster sida, sett mot vagnens gavel.
- **a.ssh\_a1** - dito i A-änden, på höger sida, sett mot gaveln.
- **a.ssh\_b0** - dito i B-änden på vänster sida, sett mot gaveln
- **a.ssh\_b1** - dito i B-änden på höger sida, sett mot gaveln.



Attachmentsen för slutsignaler ska alltså sitta så att den som heter 0 ska vara på vänstra sidan när man kollar mot gaveln i den änden och 1 på högra sidan.

(Scriptet bryr sig egentligen inte om namnet på attachment-punkterna, utan använder enbart namnen på själva mesherna för att välja vilka som ska visas och animeras osv, men vi rekommenderar att alla använder dessa namn på attachment-punkterna)

## Tillägg i configen

För att aktivera STL-koppelscriptet måste även en del saker läggas till i fordonets config:

```
script-include-table
{
    mscommonsource          <kuid2:177292:209000:5>
}
extensions
{}
```

I extensions-containern kan följande saker läggas till om det är en motor- eller manövernagn:

```
pantograph 0
engine 0
```

Dessa taggar kan ha värdena 1 eller 0 och används om det är en motor- eller manöverbagn det handlar om, annars lämnas extensions-containern tom. Pantograph anger om fordonet har en pantograph (strömavtagare) och engine talar om, om det är ett motordrivet fordon (annars en manöverbagn). Om taggen pantograph finns med (värdet har ingen betydelse) så handlar det om ett eldrivet motorvagnset och om den saknas så handlar det om ett dieseldrivet motorvagnset. Taggen pantograph har alltid ett värde större än 0 (noll) om fordonet har minst en strömavtagare.

Om värdet är 1 har fordonet en strömavtagare och pantograph "states" justeras alltid till 0-1-0-1-0 osv.

Om värdet är 2 har fordonet två strömavtagare och pantograph "states" justeras alltid till 0-1-2-0-1-2-0 osv. Dvs. båda strömavtagarna kan aldrig vara uppfällda samtidigt.

Om värdet är 3 har fordonet två strömavtagare och pantograph "states" är 0-1-2-3-0-1-2-3-0 osv. Dvs. båda strömavtagarna kan vara uppfällda samtidigt. Denna sekvens är den normala i Trainz, oavsett antalet strömavtagare. Om värdet är 0 så är det en motor- eller manöverbagn utan strömavtagare, om taggen saknas så är det en dieseldriven motor eller manöverbagn. Engine anger om det är en motorvagn (1) eller en manöverbagn(0).

Sedan så måste följande kuider in i kuid-tablen:

```
kuid-table
{
    ...
    mstrainmonitor                <kuid2:177292:204091:3>
    mscommonsource               <kuid2:177292:209000:5>
    ...
}
```

Samt någon, eller båda:

```
stl_koppel                      <kuid2:328014:599:1>
stl_koppel_ng                   <kuid2:267353:101057:1>
```

beroende på om fordonet använder normalspårs- och/eller smalspårsbiblioteket med mesher.

Dessutom måste fordonet ha ett script som ärver någon av de scriptklasser som implementerar STL-koppelfunktionaliteten. Ifall fordonet inte har ett script redan kan vi skapa ett script som heter "thisvehicle.gs". Lägg till scriptet i configfilen på följande sätt:

```
script "thisvehicle"
class "ThisVehicle"
```

## Scriptet

Fordonets scriptfil måste ärva funktionaliteten från någon av klasserna `MsVehicle` och `MsMotorVehicle` i respektive scriptfil `msvehicle.gs` och `msmotorvehicle.gs` för att STL-koppel ska fungera. Dessa två grundklasser ärvs redan av klasserna `STLVehicle`, `STLVehiclePBR`, `STLMotorVehicle`, `STLMotorVehiclePBR` och `STLSteamEngine`, så genom att låta fordonets script ärva något av dessa klasser får man automatiskt med STL-koppel. För skillnaden mellan dessa olika klasser hänvisas till dokumentet *STL Texture replacement PBR*.

Ifall fordonet inte redan har en script-fil, skapa en som heter "thisvehicle.gs" i fordonets mapp. Om det är frågan om en vagn, alltså ett fordon utan motor ska följande stå där:

```
include "msVehicle.gs"

class ThisVehicle isclass MsVehicle
{
};
```

Om det är ett lok, eller en motorvagn, med motor ska följande stå:

```
include "msMotorVehicle.gs"

class ThisVehicle isclass MsMotorVehicle
{
};
```

Om fordonet redan har ett script, öppna .gs-filen med ett textredigeringsprogram och ändra:

```
include "vehicle.gs"
till
include "msVehicle.gs"
eller om det är ett motorfordon;
include "msMotorVehicle.gs"
```

```
och
isclass Vehicle
till:
isclass MsVehicle
eller vid motorfordon:
isclass MsMotorVehicle
```

## Koppel

För koppel och kopplingsslangar till tryckluftsbronsen gäller samma sak som i gamla STL-koppel-biblioteket. I var ände på fordonet ska en attachment placeras för att sätta koppelmesherna på. I a.limfront-änden ska attachmenten heta a.koppela och i a.limback-änden a.koppelb.

För normalspårskoppel ska de placeras i med position X=0 och Z=0 och där Y är placerad i linje med buffertplåten framsida, vilket ska vara 650 mm från a.limfront/a.limback (buffertspetsen). På smalspåriga fordon ska a.koppela och a.koppelb placeras på samma

sätt, i linje med buffertplåten men här ska a.limfront/a.limback placeras 455 mm utanför dessa istället. Positiv Y-axel på attachmentsen ska peka inåt fordonet, vilket innebär att man normalt måste rotera a.koppelb 180° runt Z-axeln.

För lok och vagnar som bara har koppel i ena änden, till exempel tenderlok, tenderar och snöslungor, skippar man helt enkelt attachment och mesher i den änden som saknar koppel.

I configen ska följande två block läggas till i mesh-tablen för att lägga till koppel:

```
a_koppel
{
    mesh-asset          <KUID>
    mesh                "MESH.lm"
    anim                "MESH.kin"
    att                 "a.koppela"
    auto-create         1
    att-parent          "default"
}

b_koppel
{
    mesh-asset          <KUID>
    mesh                "MESH.lm"
    anim                "MESH.kin"
    att                 "a.koppelb"
    auto-create         1
    att-parent          "default"
}
```

Följande varianter på koppel finns att välja mellan i meshbiblioteken, se bilder på dem ovan:

KUID	MESH
<kuid2:328014:599:1>	koppel_2l
<kuid2:328014:599:1>	koppel_bo
<kuid2:267353:101057:1>	koppel_up
<kuid2:267353:101057:1>	koppel_down
<kuid2:267353:101057:1>	koppel_2chain
<kuid2:267353:101057:1>	koppel_2screw
<kuid2:267353:101057:1>	koppel_sbg

Saknar fordonet koppel i den ena änden, så skippas den delen i mesh-tablen.

## Slangkopplingar till tryckluftsbroms

Slangkopplingar för tryckluftsbroms monteras på samma attachments a.koppela och a.koppelb som själva kopplet, så inga extra attachments behövs.

För normalspåriga fordon finns slangkopplingarna i två varianter i biblioteket. Dels med blindkopplingar för slangarna, vilket användes på lok och dels i ett enklare utförande med enbart järn med en skåra i för upphängning av slangarna, vilket användes på vagnar. Det smalspåriga biblioteket innehåller i dagsläget bara en variant.

Det var vanligast att godsvagnar bara hade en kopplingsslang på ena sidan av kopplet, medan lok, personvagnar och godsvagnar som ofta kopplades in i personförande tåg hade på båda sidorna för att kunna koppla ihop slangarna på samma sida om kopplet istället för diagonalt under kopplet. Detta för att underlätta ihopkoppling då bälgar, ångvärmekopplingar och övergångsbryggor ofta var ivägen annars.

För att lägga till tryckluftsslangar, lägg till följande i configens mesh-table:

```
a_luft_0
{
    mesh-asset          <KUID>
    mesh                 "MESH.lm"
    anim                 "MESH.kin"
    att                  "a.koppela"
    auto-create          1
    att-parent           "default"
}

a_luft_1
{
    mesh-asset          <KUID>
    mesh                 "MESH.lm"
    anim                 "MESH.kin"
    att                  "a.koppela"
    auto-create          1
    att-parent           "default"
}

b_luft_0
{
    mesh-asset          <KUID>
    mesh                 "MESH.lm"
    anim                 "MESH.kin"
    att                  "a.koppelb"
    auto-create          1
    att-parent           "default"
}

b_luft_1
{
    mesh-asset          <KUID>
    mesh                 "MESH.lm"
```

```

    anim                "MESH.kin"
    att                  "a.koppelb"
    auto-create          1
    att-parent            "default"
}

```

Byt ut kuid samt namnet på meshen mot någon av följande:

KUID	MESH X_luft_0	MESH X_luft_1
<kuid2:328014:599:1>	luft_0_vagn	luft_1_vagn
<kuid2:328014:599:1>	luft_0_lok	luft_1_lok
<kuid2:267353:101057:1>	luft_0_anim	luft_1_anim

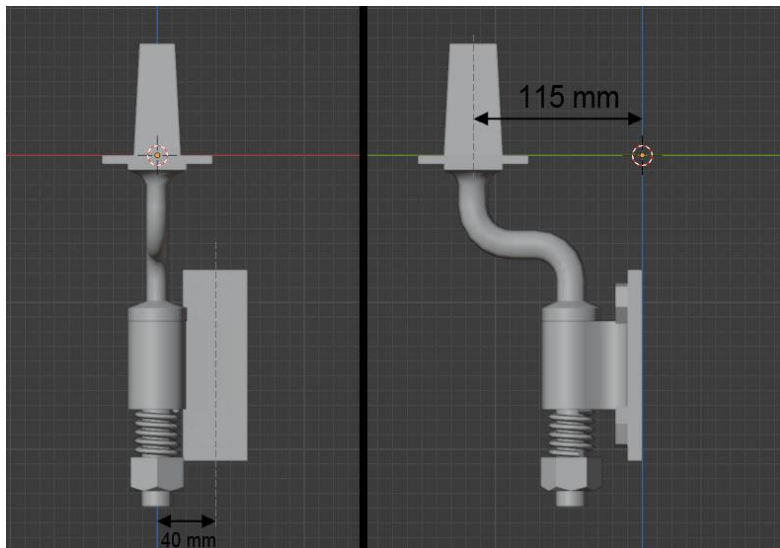
På godsvagnar och andra fordon med enbart en slangkoppling, ta bort de två mesher som heter a\_luft\_0 och b\_luft\_0, så att bara kopplingen på höger sida om kopplet (sett mot gaveln på vagnen) är kvar.

Saknas koppel i ena änden skippar man helt enkelt båda dessa containers i den änden.

Fordon med Scharfenbergkoppel har normalt inga tryckluftslangar, eller har dessa inbyggda i kopplet, så då skippar man dessa helt.

## Slutsignaler

Slutsignaler och hållarna till dessa monteras på attachments som heter a.ssh\_a0, a.ssh\_a1, a.ssh\_b0 och a.ssh\_b1. De som heter "a" ska placeras i a.limfront-änden på vagnen och de som heter "b" i a.limback-änden. Placeringen av attachmentsen för dessa skiljer sig mot det gamla meshbiblioteken. I det nya biblioteket ska de placeras på ytan på vagnsidan där signalhållaren ska sitta på. I höjddled kommer botten på slutsignalen (och plattan på slutsignalhållaren) att hamna i höjd med attachment-punkten. Hållarna är asymmetriska genom att fästplattan sitter 40 mm förskjuten åt ena eller andra hållet, så meshen finns i två varianter. Attachmenten är centrerad till slutsignalen, inte till fästplattan, vilket innebär att man måste flytta attachmenten 40 mm i vagnens längdled för att få fästplattan centrerad där man vill ha den. Normalt sett ska den flyttas i Y-led mot vagnens mitt till. Se bilden nedan för slutsignalens position i förhållande till attachment-punkten.



För vagnar som enbart har ett par slutsignalhållare i vagnens ena ände, behöver enbart attachments för den änden sättas ut. Man behöver dock fortfarande ha mesher för slutsignalerna för "båda ändar" av vagnen och de placeras då på samma hållare och attachment. Detta krävs dels av scriptet för att det ska fungera och dels för att slutsignallykta har olika färg på ljuset från lyktan åt olika håll. På vagnar som saknar slutsignalhållare, skippar man helt enkelt attachments och mesher för dessa.

För slutsignaler kan man välja om man vill ha skärmar, lyktor, eller båda varianterna. Väljer man att ha båda så kommer dessa automatiskt att bytas beroende på tid på dagen. Väljer man bara en typ så kommer den typen att visas dygnet runt. Signalerna byts mellan dagsignaler och nattsignaler enbart när nya signaler sätts ut i samband med att någon vagn kopplas in eller ut ur tåget.

För slutsignalhållarna, lägg till följande block i mesh-tablen:

```
ssh_a0
{
  mesh-asset                                <kuid2:328014:599:1>
  mesh                                       "ssh_h.lm"
  att                                       "a.ssh_a0"
  auto-create                               1
  att-parent                               "default"
}

ssh_a1
{
  mesh-asset                                <kuid2:328014:599:1>
  mesh                                       "ssh_v.lm"
  att                                       "a.ssh_a1"
  auto-create                               1
  att-parent                               "default"
}

ssh_b0
{
```

```

        mesh-asset          <kuid2:328014:599:1>
        mesh                "ssh_h.lm"
        att                 "a.ssh_b0"
        auto-create         1
        att-parent          "default"
    }

    ssh_b1
    {
        mesh-asset          <kuid2:328014:599:1>
        mesh                "ssh_v.lm"
        att                 "a.ssh_b1"
        auto-create         1
        att-parent          "default"
    }

```

På vagnar med bara ett par slutsignalhållare i ena änden av vagnen, ta bort mesherna till de två hållare som inte används. Ändra sedan attachment-punkten för de slutsignalskärmar och -lyktor i den änden som *inte* har hållarna kvar, till motsvarande för andra änden (a eller b) och byt sedan sida (0 eller 1) på dem. Till exempel om slutsignalhållare bara finns i B-änden, byt attachment-punkten på signalerna och lyktorna i A-änden från a.ssh\_a0 till a.ssh\_b1 och a.ssh\_a1 till a.ssh\_b0. Man behöver byta sida på dem för att lyktornas färgade ljus ska lysa åt rätt håll.

För slutsignalskärmar, lägg till följande block i mesh-tablen:

```

a_slutsignal_0
{
    mesh-asset          <kuid2:328014:599:1>
    mesh                "MESH.lm"
    att                 "a.ssh_a0"
    auto-create         0
    att-parent          "default"
}

a_slutsignal_1
{
    mesh-asset          <kuid2:328014:599:1>
    mesh                "MESH.lm"
    att                 "a.ssh_a1"
    auto-create         0
    att-parent          "default"
}

b_slutsignal_0
{
    mesh-asset          <kuid2:328014:599:1>
    mesh                "MESH.lm"
    att                 "a.ssh_b0"
    auto-create         0
    att-parent          "default"
}

```

```

}

b_slutsignal_1
{
    mesh-asset                <kuid2:328014:599:1>
    mesh                      "MESH.lm"
    att                       "a.ssh_b1"
    auto-create               0
    att-parent                "default"
}

```

KUID	MESH
<kuid2:328014:599:1>	skarm_gul
<kuid2:328014:599:1>	skarm_vit

Det finns två mesher, dels i röd/vit färgsättning vilken användes fram till ca 1940 och dels gul/röd färgsättning som användes efter det.

För slutsignallyktor lägg till följande i mesh-tablen:

```

a_slutlykta_0
{
    mesh-asset                <kuid2:328014:599:1>
    mesh                      "lykta_1930t_v.lm"
    att                       "a.ssh_a0"
    auto-create               0
    att-parent                "default"
}

a_slutlykta_1
{
    mesh-asset                <kuid2:328014:599:1>
    mesh                      "lykta_1930t_h.lm"
    att                       "a.ssh_a1"
    auto-create               0
    att-parent                "default"
}

b_slutlykta_0
{
    mesh-asset                <kuid2:328014:599:1>
    mesh                      "lykta_1930t_v.lm"
    att                       "a.ssh_b0"
    auto-create               0
    att-parent                "default"
}

```

```

b_slutlykta_1
{
    mesh-asset                <kuid2:328014:599:1>
    mesh                      "lykta_1930t_h.lm"
    att                       "a.ssh_b1"
    auto-create               0
    att-parent                "default"
}

```

Notera att till skillnad från den motsvarande lykt-varianten i det gamla STL-koppel-biblioteket, behöver denna nya version inte ha corona effekterna. Själva ljuset är istället löst med emission-mappen i PBR-texturen. Detta betyder att problemen med att coronan syns igenom väggar som kunde uppstå ibland med den gamla varianten nu är borta.

### Inbyggda slutsignaler

Scriptet har även stöd för inbyggda slutsignaler, vilket fungerar precis som tidigare i STL-koppel. Detta används till exempel på vissa ellok för att markera B-änden, eller som inbyggda slutsignaler på de personvagnar som levererats efter 60-talet.

Dessa ljus sköts genom corona-effekter. Lägg då till följande i effects-blocket på defaultmeshen i mesh-table:

```

effects
{
    a_slutljus_0
    {
        kind                "corona"
        att                  "a.XXX"
        directional          1
        texture-kuid         <KUID>
    }
    b_slutljus_0
    {
        kind                "corona"
        att                  "a.XXX"
        directional          1
        texture-kuid         <KUID>
    }
}

```

Här behöver man lägga till egna attachmentspunkter, förslagsvis med namn a.slutljus\_a0, a.slutljus\_b0 osv.

Vill man istället för corona-effekter använda det mer moderna sättet med emission i en PBR-texture istället, kan man använda slutlykta-containrarna (a\_slutlykta\_0, ...) istället, och bara skapa en mesh med slutsignalens lins som ett emissive PBR-material.

## Övergångsbryggor och dragspelsbälgar

STL-koppel har sedan tidigare stöd för animerade övergångsbryggor och bälgar mellan fordon. Och funktionen på dessa är exakt som tidigare. Meshbiblioteken innehåller inga bryggor eller bälgar, utan det är upp till byggaren att skapa sådana som passar fordonet, eller att skapa ett eget bibliotek med sådana. Dessa kan antingen fästas i a.koppela och a.koppelb, eller i en helt egen attachment.

För övergångsbryggor kan göras på två olika sätt:

1. Vanlig animation med en .kin-fil över valfritt antal frames där första framen är uppfälld och sista framen är utfälld brygga. Animationen körs baklänges när bryggan fälls upp.
2. On/off-metoden, där man skapar två statiska mesher som scriptet byter emellan. On är utfälld brygga och off är uppfälld.

Bryggor med animation läggs till i mesh-tablen på fordonet på följande sätt:

```
a_brygga
{
    mesh                "MESH.lm"
    anim                "MESH.kin"
    auto-create         1
    att                 "a.XXX"
}
b_brygga
{
    mesh                "MESH.lm"
    anim                "MESH.kin"
    auto-create         1
    att                 "a.XXX"
}
```

Bryggor gjorda med on/off-metoden läggs istället till på följande sätt i mesh-tablen:

```
a_brygga_on
{
    mesh                "MESH_on.lm"
    auto-create         0
    att                 "a.XXX"
}
a_brygga_off
{
    mesh                "MESH_off.lm"
    auto-create         1
    att                 "a.XXX"
}
b_brygga_on
{
    mesh                "MESH_on.lm"
    auto-create         0
    att                 "a.XXX"
}
```

```

}
b_brygga_off
{
    mesh                "MESH_off.lm"
    auto-create         1
    att                 "a.XXX"
}

```

On-meshen är utfälld brygga och off är infälld brygga.

Saknas övergångsbryggor i ena änden av fordonet, skippar man helt enkelt dessa i mesh-tablen.

Övergångsbälgar har en övergångsbrygga inuti, så den bryggan skapas på sättet som beskrivits ovan. Bälgen skapas sedan som en separat mesh med en separat animation. Detta för att övergångsbryggan ska kunna fällas ut oberoende av bälgen när man kopplar vagnen till en vagn utan bälgar.

Bälg-meshen kan skapas på tre olika sätt:

1. Vanlig animation med en .kin-fil över valfritt antal frames, där frame 1 är infälld bälg och sista framen är utfälld bälg. Animationen körs baklänges vid infällning.
2. On/off-metoden, med en statisk mesh för infällt läge och en statisk mesh för utfällt läge som scriptet byter mellan.
3. En "stop-motion" animation med flera statiska mesher med stegen mellan helt infälld och helt utfälld, där scriptet byter mellan dem för att få det att se ut som en animation. Detta sätt härstammar från tiden då skalning inte var möjligt med Blender-exportern, vilket gjorde det svårt att skapa snygga bälg-animationer med .kin-filer. Det är rekommenderat att använda vanliga .kin-filer istället. Det här sättet finns dock kvar i scriptet för bakåtkompatibilitet och är beskrivet närmare i den äldre guiden till STL-koppel.

Har man en vulstbälg kan man göra på två olika sätt beroende på om man vill att dragspelsbälgar ska kopplas till den eller inte. Om man *inte* vill att dragspelsbälgar ska kopplas med dem, gör man bälgen som en del av korgen på vagnen och bara animerar bryggorna. Då ser scriptet vagnen som en vagn med enbart övergångsbryggor. Om man vill att andra dragspelsbälgvagnar ska fälla ut bälgen använder man on/off-metoden, men har båda mesherna likadana, eftersom vulstbälgar inte rör på sig. Då kommer dragspelsbälgen att tro att vagnen den kopplar mot har en utfällbar bälg och därför fäller den ut sin. Vilken av dessa varianter man väljer kommer inte att göra någon skillnad när man kopplar två vulstbälgvagnar mot varandra.

Scriptet stödjer även alla kombinationer att man har olika varianter i olika ändar av vagnen, exempelvis: brygga i bara ena änden; brygga i ena och brygga och bälg i den andra änden; brygga och bälg i ena änden och ingenting i den andra änden osv..

För bälgar animerade med .kin-filer, lägg till dem i mesh-tablen på följande sätt:

```

a_balga
{
    mesh                "MESH.lm"
    anim                "MESH.kin"
    auto-create         1
    att                 "a.XXX"
}
b_balga
{
    mesh                "MESH.lm"
    anim                "MESH.kin"
    auto-create         1
    att                 "a.XXX"
}

```

För bälgar med on/off-metoden lägg till följande i mesh-tablen:

```

a_balga_on
{
    mesh                "MESH_on.lm"
    auto-create         0
    att                 "a.XXX"
}
a_balga_off
{
    mesh                "MESH_off.lm"
    auto-create         1
    att                 "a.XXX"
}
b_balga_on
{
    mesh                "MESH_on.lm"
    auto-create         0
    att                 "a.XXX"
}
b_balga_off
{
    mesh                "MESH_off.lm"
    auto-create         1
    att                 "a.XXX"
}

```

On-mesherna är utfälld bälg och off är infälld bälg.

Bälgvagnar ska dessutom ha någon form av övergångsbrygga enligt instruktionerna i ovan.

För den gamla "stop-motion"-varianten på animation, hänvisas till gamla manualen.

## Gaveldörrar

Scriptet har stöd för att automatiskt öppna gaveldörrar när en personvagn stannar vid en station. Inga standardmesher finns för detta i meshbiblioteken, utan de måste skapas för

varje enskild vagn. Gaveldörrarna animeras och exporteras som två separata mesher och sätts fast på en egen attachment, eller på a.koppela/b. Animeringen görs på valfritt antal frames, 30 rekommenderas, där frame 1 är stängd dörr och högsta framevärdet är öppen dörr. Scriptet ser hur många frames det är och spelar animationen baklänges när dörren stängs. Lägg till följande i configen i mesh-tablen:

```
a_gabledoor
{
    mesh                "XXX.lm"
    anim                "XXX.kin"
    att                 "a.XXX"
    auto-create         1
}

b_gabledoor
{
    mesh                "XXX.lm"
    anim                "XXX.kin"
    att                 "a.XXX"
    auto-create         1
}
```

Byt ut meshnamn och animationsfilnamn och namn på attachmentpoint till de som är aktuella.

## Att byta koppeltyp

Eftersom alla koppel, förutom Scharfenbergkopplet, har en standardlängd så kan man enkelt byta koppeltyp på ett lok eller en vagn genom att byta mesh- och animationsnamnet på kopplen till de man vill ha. Detta är framförallt intressant för smalspårskoppel, eftersom olika banor använde olika modeller på koppel.

## Att göra egna mesher

I de andra flesta fall fungerar det att använda de koppel och tryckluftsslangar som finns i biblioteken, men i vissa fall vill man skapa egna modeller och animationer på sådana. Den byggare som vill skapa egna mesher till koppel eller slutsignaler kan därför fritt göra det. Använd då samma namn i mesh-tablen på mesherna, men peka ut din mesh istället för mesherna i meshbibliotken som beskrivs ovan.

För att delarna ska fungera med scriptet måste de animeras på rätt sätt. För koppel gäller följande: Animationen får vara över valfritt antal frames och ska ha 5 poser. Två av dem är samma för att det ska gå att animera direkt mellan alla poser utan att passera en annan pose. Normalspårskopplet som är med i *Svenska koppel och slutsignaler PBR* är animerat över 240 frames.

Om X är antalet frames i animationen så är poserna följande:

1. Frame 1: Upphängd på upphängningskroken, eller säkerhetsdragkroken under koppelkroken. Används när fordonet är ihopkopplad med ett annat fordon, men det andra fordonets koppel används
2. Frame X/4: Hängande rakt ned. Används på koppel på vagnar när de är uppställda och ej är inkopplade i ett tåg, eller är ihopkopplade med en annan vagn.
3. Frame (X + 1)/2: Kopplad på andra fordonets dragkrok. Används när fordonet är ihopkopplat med ett annat fordon
4. Frame (X \* 3)/4: På egen dragkrok. Används på främsta och bakersta kopplet i ett tåg
5. Frame X: Hängande rakt ned (samma pose som nummer 2)

För smalspårskoppel får man tolka betydelsen av dessa poser och hitta motsvarande för den typen av koppel. För tvåkammarkoppel och andra koppel där båda sidor ska kopplas ihop på ett symmetriskt sätt kommer antagligen pose 3 och 4 vara samma. För enkammarkoppel, liksom normalspårskopplet kommer de att vara olika.

För vänster (sett mot vagnens gavel) tryckluftsslang, luft\_0, alltså den som enbart finns på vagnar som går i persontåg, gäller följande:

1. Frame 1: Ej kopplad, upphängd i sin hållare
2. Frame X: Kopplad med motsatt vagn, rakt fram i Y-led (ej diagonalt under kopplet)

För höger tryckluftsslang (som alla vagnar med tryckluftsbromsledning har), luft\_1, gäller följande:

1. Frame 1: Kopplad med motsatt vagn, diagonalt under kopplet.
2. Frame X/2: Ej kopplad, upphängd i sin hållare.
3. Frame X: Kopplad med motsatt vagn, rakt fram i Y-led

Luftslangarna som är med i *Svenska koppel och slutsignaler PBR* är animerade över 60 (luft\_0) respektive 120 (luft\_1) frames.

På mesherna för tryckluftslangarna i normalspårskoppelbiblioteket är korsningspunkten mellan kopplingsnävarna på följande position relativt origo för koppling rakt:

$X = \pm 480\text{mm}$ ,  $Y = -650\text{mm}$ ,  $Z = 370\text{mm}$

För diagonal koppling under kopplet är korsningspunkten:

$X = 0\text{mm}$ ,  $Y = -650\text{mm}$ ,  $Z = 650\text{mm}$

För smalspårskopplen är motsvarande mått för kopplingsnävarnas korsningspunkt vid rak koppling:

$X = 414.3\text{mm}$ ,  $Y = -455\text{mm}$ ,  $Z = 325\text{mm}$

Och för diagonal koppling under kopplet:

$X = 0\text{mm}$ ,  $Y = -455\text{mm}$ ,  $Z = 450\text{mm}$

Övriga mesher som är med i mesh-biblioteken är statiska och går enkelt att ersätta. Se bara till att matcha slutsignalernas placering med själva slutsignalen. Se illustrationen tidigare i detta dokument.

Mycket nöje!

// Korvtiger 2023-09-14