

Push-Pull Trains

The following is an example for a simple Push-Pull train; the train consists of an engine and a number of passenger coaches. The last coach will become a cab control coach automatically. There is no need to have a separate cab coach by vehicle ID. Both vehicles must be present in this GRF, some sprites and code cannot reside in another GRF.

In this example the train is built engine leading.

It is possible to have and start a train with a cab coach or engine; but extra code will be required.

Rules :

- . You cannot have more than 1 engine in a train.
- . The first and last vehicle must have the same length (there are exceptions though)
- . All vehicles in the middle of a train must be the same (exceptions possible, but extra code required)
- . It is possible to have different engines, but each engine can only have one kind of coach attached (exceptions possible with extra code)
- . All real sprites (engine and coaches) must reside above action-3 and code logic for engine.

Requirements :

- . The engine

 Action-0 properties must include :

 Callback flags (0x1E) - value 1 for powered wagons and visual effects

 Visual effects and wagon power (0x22) - set to 0x40 [no visual effects, visual effects will be set via callback]

 Graphics :

 2 sets of sprites for engine leading (headlights on) and engine trailing (taillights on)

- . The passenger coaches

 Action-0 properties must include

 Callback flags (0x1E) - value 1 for powered wagons and visual effects

 Visual effects and wagon power (0x22) - set to 0x40 [no visual effects, visual effects will be set via callback]

 Graphics :

 3 sets of sprites for first wagon (cab coach) when leading (with headlights), middle wagons and last wagon (cab coach) when trailing (with taillights)

PushPull[1].nfo

NFO code for cab coach including middle coaches

```
Action-1 for coaches
    00 - coach leading
    01 - middle coach
    02 - coach trailing

-1 * 9      02 00 B0 01 01 00 00 00 00
-1 * 9      02 00 B1 01 01 01 00 01 00
-1 * 9      02 00 B2 01 01 02 00 02 00

-1 * 14     02 00 AE 81 40 08 FF 01 B2 00 00 00 B1 00
            // if last vehicle, then show cab coach
trailing, otherwise it is a middle car
-1 * 14     02 00 AF 81 40 00 FF 01 B0 00 00 00 AE 00
            // if first vehicle, then show cab coach
leading
-1 * 14     02 00 F0 81 0C 00 FF 01 40 80 10 10 AF 00
            // CB 10 turn off visual effects for wagon
```

NFO code for engine

```
Action-1 for engine
    00 - engine leading
    01 - engine trailing

-1 * 9      02 00 B0 01 01 00 00 00 00
-1 * 9      02 00 B1 01 01 01 00 01 00

-1 * 14     02 00 AF 81 40 00 FF 01 B0 00 00 00 B1 00
            // if first vehicle, then show engine leading,
otherwise engine trailing
-1 * 14     02 00 F1 81 0C 00 FF 01 38 80 10 10 AF 00
            // CB 10 turn on visual effects (electric sparks) for
engine
```

NFO code Push-Pull logic for wagon

```
-1 * 14     02 00 BE 81 40 08 FF 01 F1 00 00 00 F0 00
            // if train travels in reverse, goto engine for last
vehicle, otherwise it's a wagon
-1 * 14     02 00 BF 82 FF 00 01 01 F0 00 00 00 BE 00
            // if train travels normally, goto wagon
```

NFO code Push Pull logic for engine

```
-1 * 14     02 00 AD 82 FF 00 01 01 F1 00 00 00 F0 00
            // if train travels normally, goto engine, otherwise
goto wagon
```

NFO code for engine

```
-1 * 14     02 00 AE 81 C6 00 FF 01 FF 80 WW[id] WW[id]
FD 80      // allow passenger coaches only
-1 * 14     02 00 AF 81 0C 00 FF 01 AE 00 1D 1D AD 00
            // callback sorter : can wagon be attached
```

PushPull[1].nfo

NFO code Action-3 entry for engine

```
-1 * 10      03 00 01 EE[id] ... .. AF 00  
           // Engine entry action-3
```

```
-1 * 7       03 00 81 WW[id] 00 BF 00  
           // Wagon override for passenger coaches, goto  
push-pull logic for wagon
```

Note: other desirable code has been omitted; e.g. purchase list entries, refitting, correct train building etc.

