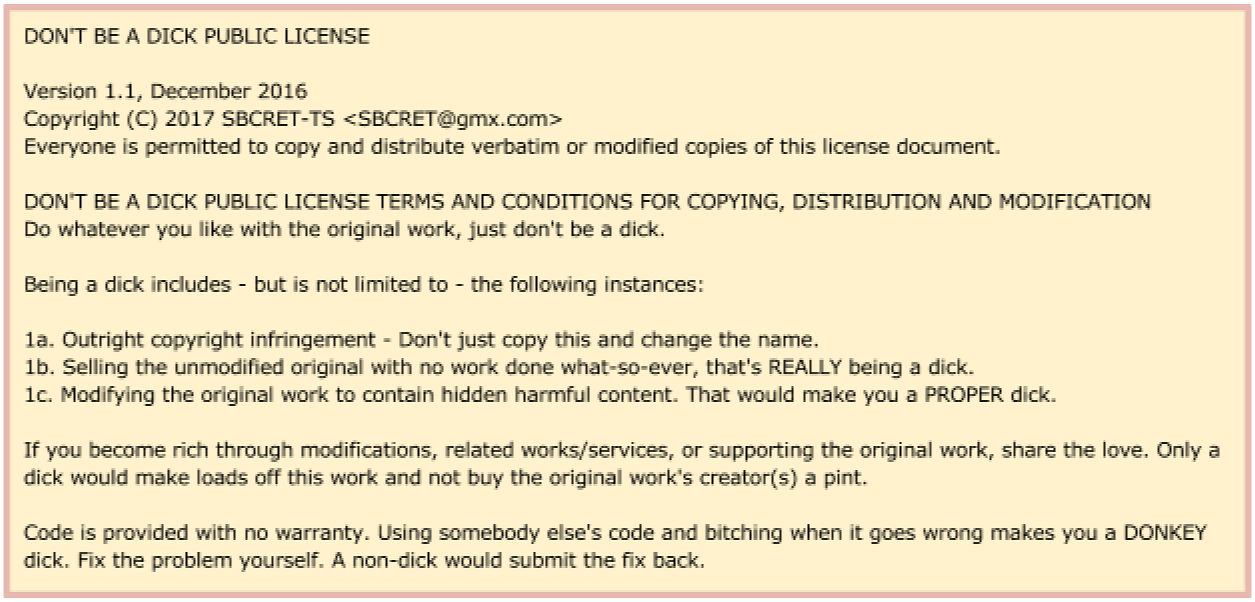
TrainTrax3U
User Guide

Auxiliary Information

Document Edition

A17E11R02, by MinecraftSBC

Licensing information



DON'T BE A DICK PUBLIC LICENSE

Version 1.1, December 2016
Copyright (C) 2017 SBCRET-TS <SBCRET@gmx.com>
Everyone is permitted to copy and distribute verbatim or modified copies of this license document.

DON'T BE A DICK PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
Do whatever you like with the original work, just don't be a dick.

Being a dick includes - but is not limited to - the following instances:

1a. Outright copyright infringement - Don't just copy this and change the name.
1b. Selling the unmodified original with no work done what-so-ever, that's REALLY being a dick.
1c. Modifying the original work to contain hidden harmful content. That would make you a PROPER dick.

If you become rich through modifications, related works/services, or supporting the original work, share the love. Only a dick would make loads off this work and not buy the original work's creator(s) a pint.

Code is provided with no warranty. Using somebody else's code and bitching when it goes wrong makes you a DONKEY dick. Fix the problem yourself. A non-dick would submit the fix back.

“TrainTrax”, “TrainTrax3U” and traintrax are unregistered trademarks (™) of SBCRET-TS.

Table of Contents

Disclaimer (SBCRET Advanced Docs)

Nothing written in this document can be understood by **pathetic, vacuous** humans without the supervision of our ***superior*** SBCRET “engineers”.

DO NOT ATTEMPT to read this document if you have insufficient knowledge in English language or lacks basic reasoning.

SBCRET is **NOT RESPONSIBLE** for any (including but not limited to)

- Increase in alcohol consumption
- Head banging
- Frustration
- Depression
- Increased risk for cardiac complications
- High blood pressure

as a result of reading this document.

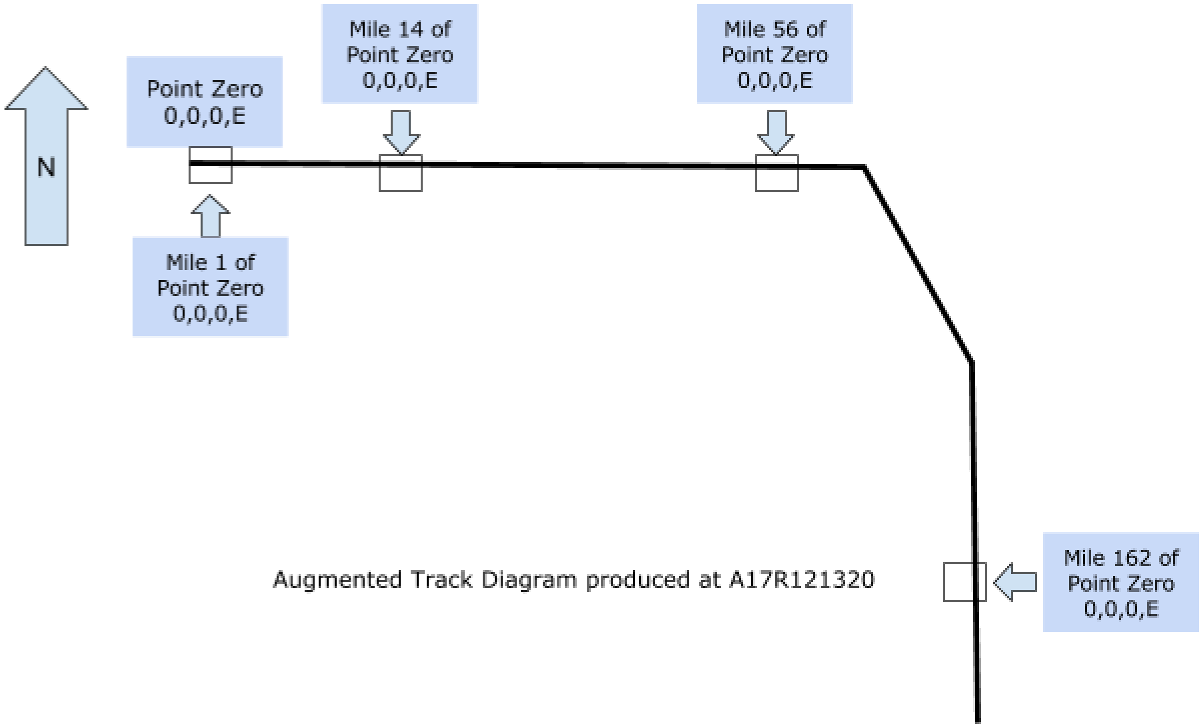
Skills required

- Basic programming knowledge
- Ability to use commands
- English language
- Basic reasoning and logic

Glossary

Point Zero (P0) refers to a reference starting point used in the Scanning Process.

Mileage (Mile) refers to a specific meter of track of distance (in meters) starting from the specified Point Zero (P0).



Parameters (Param) refers to the details of a TrainTrax Component, including the details of Point Zero (P0).

Raw Dump (Dump) refers to the results produced by a TrainTrax function.

Identifier (Ident / ID) refers to the name of the specific TrainTrax Component.

Main Feed (MF) refers to a merged collection of Point Zero Raw Dumps.

Data type / Notations

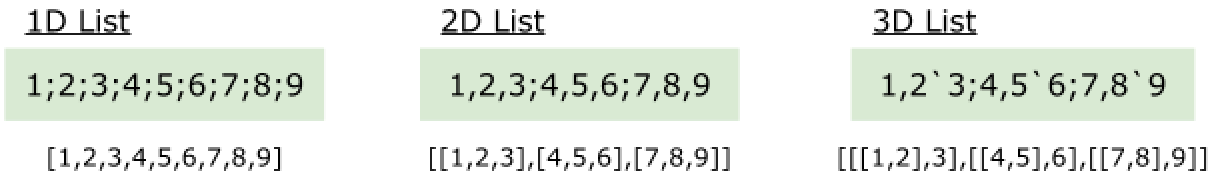
Variable Storage

TrainTrax3U uses *strings* formatted in a certain way to store all its data.

String is used for the following reasons:

- Reduces type confusion between modules
- Reduces index confusion before and after modification
  *P.S. Skript list variables DON'T ****ING UPDATE INDEXES AFTER MODIFICATION!!! :(*
- Ability to be stored in a list variable
  *P.S. Skript DON'T ****ING ALLOW 2D LISTS to be stored (actually yes but like ::1::2)*
- Easy modifications using join / split

List Notation (SDC List Strings)



1D List

1;2;3;4;5;6;7;8;9

[1,2,3,4,5,6,7,8,9]

2D List

1,2,3;4,5,6;7,8,9

[[1,2,3],[4,5,6],[7,8,9]]

3D List

1,2`3;4,5`6;7,8`9

[[[1,2],3],[[4,5],6],[[7,8],9]]

Each *outermost layer* element is separated by a *semicolon*: “;”
Each *innermost layer* element is separated by a *comma*: “,”
Other delimiters used are listed below (in order):

[“`” , ”|” , “~”, ”\” , ”/”]

Each list is **null-terminated**.

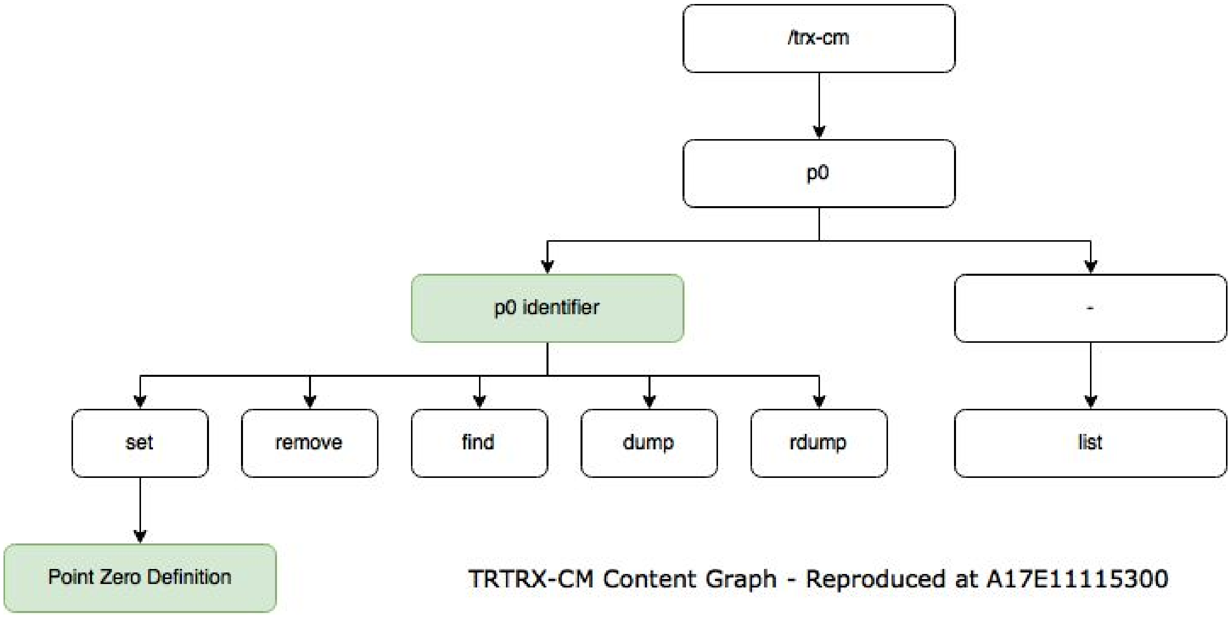
List should **only** contain the following data types:

[integer , number , string , text]

And **must not** contain the following characters:

[“:” , “;” , “,” , “`” , ”|” , ”~” , ”\” , ”/”]

Content Graph



TRTRX-CM Content Graph - Reproduced at A17E11115300

Green elements indicate evaluation elements

Sample valid command with reference to the content graph above:

- /trx-cm p0 nsl_dn rdump
- /trx-cm p0 nsl_dn set 179 43 697 n 2074

Each content graph represents **all possible options of a command**.

Each layer of a content graph indicate **one argument**, with the first layer as the command itself, and the last layer as the final argument.

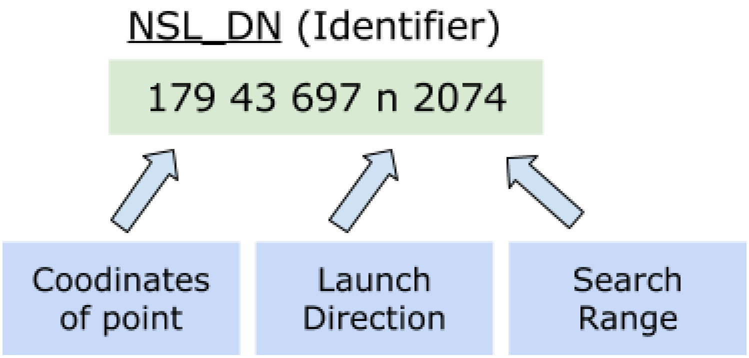
Evaluation elements refer to arguments to be replaced with **custom data**.

Unless otherwise specified, **the last element must be reached** to be deemed as a valid command. Only the **last element** in the content graph can include the “ ” character.

TrainTrax Components

Point Zero

Declaration

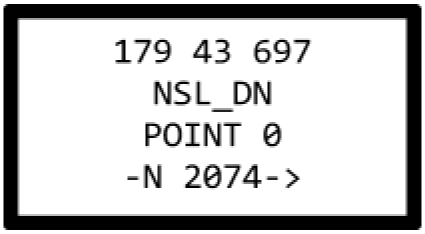


The declaration of a Point Zero consists of -

[0:3]	[X, Y,Z] coordinate of the starting point
[3]	Launch direction (n/e/w/s)
[4]	Scan range (Mile searched to trigger the SIGKILL Exit)

Marking

Users are advised to mark named Point Zero(s) with formatted signs.
The sign should include all parameters and the identifier.

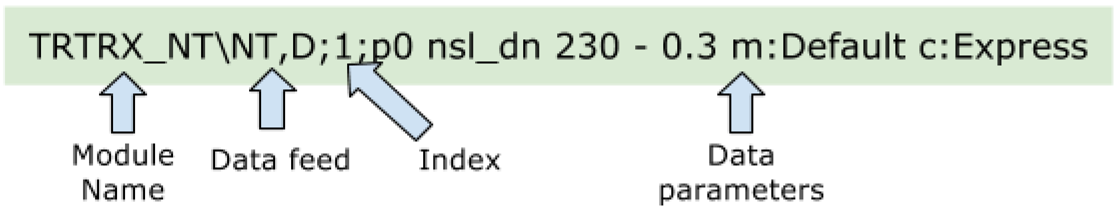
Sample Placement	*Sample Marker*
	 Line 1: X, Y,Z Coordinate Line 2: Point Zero identifier Line 3: “POINT 0 / POINT ZERO” Line 4: Direction and Scan Range *(with arrow indicating direction)*

Train data signs (Handled by SigNelson)

Declaration

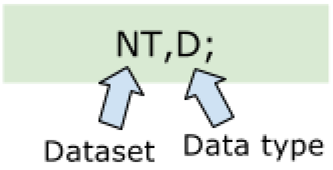
Next Train Information

123,65,782 (Identifier)



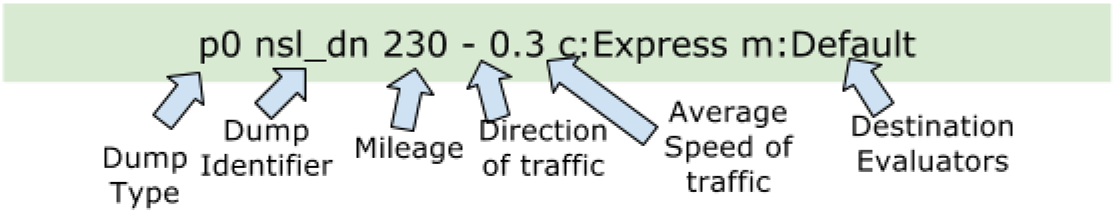
```
TRTRX_NT\NT,D;1;p0 nsl_dn 230 - 0.3 m:Default c:Express
```

Data Feed



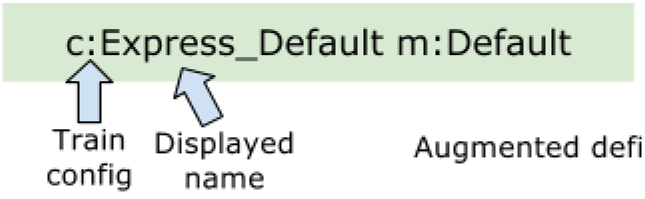
```
NT,D;
```

Data Parameters



```
p0 nsl_dn 230 - 0.3 c:Express m:Default
```

Destination Evaluators



```
c:Express_Default m:Default
```

Augmented definition produced at A17E10221200

The declaration of a Next Train Information sign consists of -

[0]	Module Name (TRTRX_NT: TrainTrax3U Next Train)
[1]	Data feed
[2]	Index (Index to use for the data feed, starting from 1)
[3]	Data parameters

The declaration of a Data Feed consists of -

[0]	Dataset (NT: Next Train)
[1]	Data Type (D: Destination, E: ETA)

The declaration of a Data Parameter consists of -

[0]	Dump Type (p0: Point Zero)
[1]	Dump Identifier / Component Identifier
[2]	Mileage
[3]	Direction of traffic (Direction in which the traffic is coming from) (+ / -)
[4]	Average speed of traffic (in m/tick)
[5]	Destination Evaluators

The declaration of a Destination Evaluator consists of -

[0]	Train configuration
[1]	Displayed name ("_" is replaced with spaces upon evaluation)

Caution: Destination evaluator is evaluated from LEFT to RIGHT, using the first matching train configuration.

For available cart types, refer to
TrainTrax3U Core: Function FindTrain / FindTrainS: Output: Detected Trains

For more information on setup, modification, removal and viewing of sign data,
refer to SigNelson User Guide.

TrainTrax3U Core

Summary

TrainTrax3U Core - Point Zero and Mile Dump
Author: MinecraftSBC, blackon108 (Algorithm advisory)

This module is responsible for train tracking on a single continuous track, and for obtaining track data.

This module contains 3 functions -

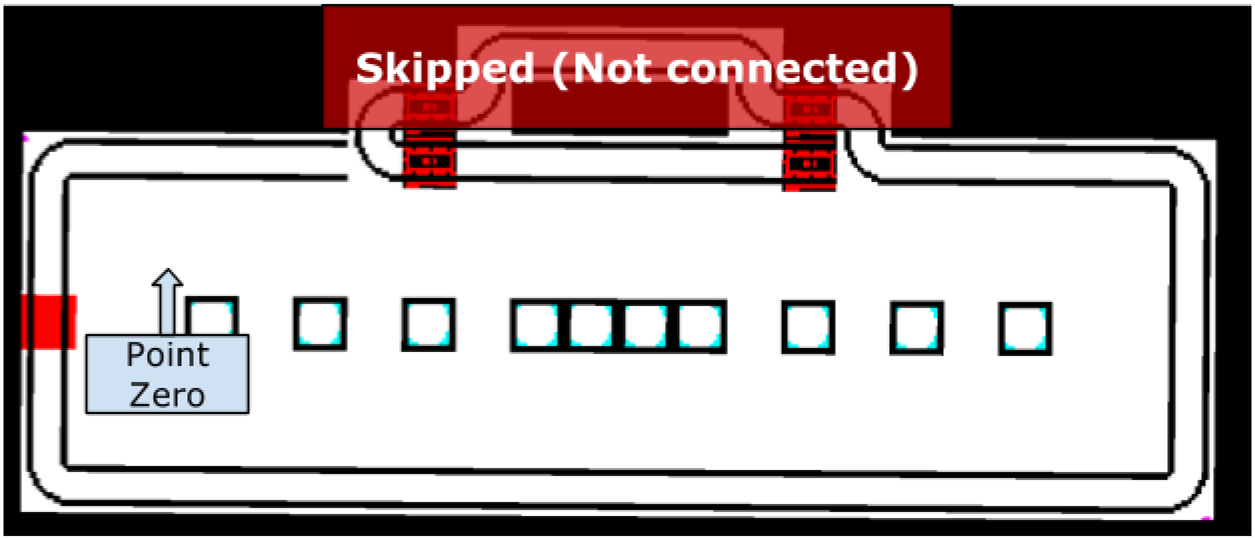
- FindTrain(x: integer, y: integer, z: integer, dir: string, itin: integer) :: string:
- FindMile(x: integer, y: integer, z: integer, dir: string, itin: integer) :: string:
- FindTrainS(x: string) :: string:

Track Scanning Mechanics

Summary

Track scanning starts at specified Point Zero then continue on along the track (analogy: virtual minecart).

Caution: "Disconnected" tracks (unreachable by minecart path finding) CANNOT be scanned.



Termination conditions

The scanning process will be automatically terminated under one of the conditions listed.

SIGKILL EXIT: The scanner has reached the termination mile specified by the Point Zero.

DUPLOC EXIT: The scanner has reached the Point Zero for a second time.

EOL EXIT: The scanner has reached the end of a track.

Train Scanning Mechanics

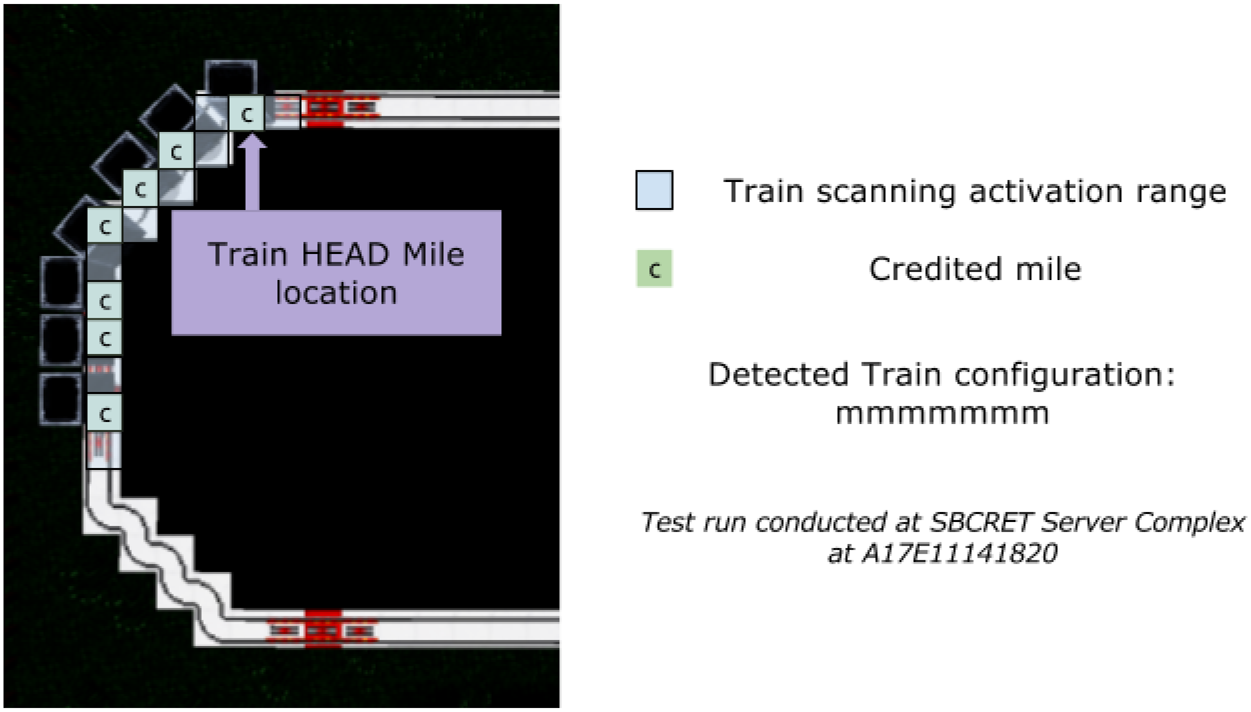
Summary

Train scanning starts when a minecart is within a 1.5m radius of a mile.

Minecarts <=1.5m apart is accounted towards the same train.
A car is recorded if the center of the minecart is within a 0.665m radius of the center of a mile.
(Using blackon108's half-meter compensation algorithm)

Train scanning stops when no minecart is detected within a 1.5m radius of a mile, or it has reached one of the track scanning exit conditions.
For exit conditions, refer to [TrainTrax3U Core: Track Scanning Mechanics: Termination Conditions]()

Train head mileage is assumed as 1 block before the train scanning process terminates.


c
c
c
c
c
c
c
Train HEAD Mile location
Train scanning activation range
c
Credited mile
Detected Train configuration:
mmmmmmm

Test run conducted at SBCRET Server Complex at A17E11141820

Function FindTrain / FindTrainS

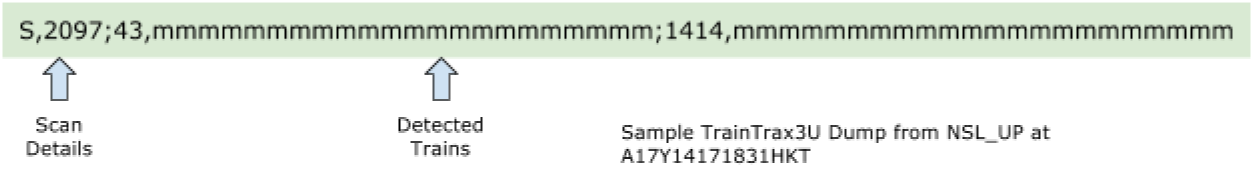
Input

This function takes a Point Zero declaration, as individual params.
FindTrain(x: integer, y: integer, z: integer, dir: string, itin: integer) :: string:

This function takes a Point Zero declaration, as a continuous string.
FindTrainS(x: string) :: string:

Output

The function produces the following result.



```
S,2097;43,mmmmmmmmmmmmmmmmmmmmmm;1414,mmmmmmmmmmmmmmmmmmmmmm
```

Sample TrainTrax3U Dump from NSL_UP at
A17Y14171831HKT

Scan Details

[0]	Exit condition S: SIGKILL Exit D: Duploc Exit E: EOT Exit
[1]	Miles scanned.

Detected Trains

[0]	Train location (**Head** mileage)
[1]	Train configuration (tail to head) M: regular minecart C: chested minecart H: hopper minecart F: furnace minecart T: TNT minecart

Function FindMile

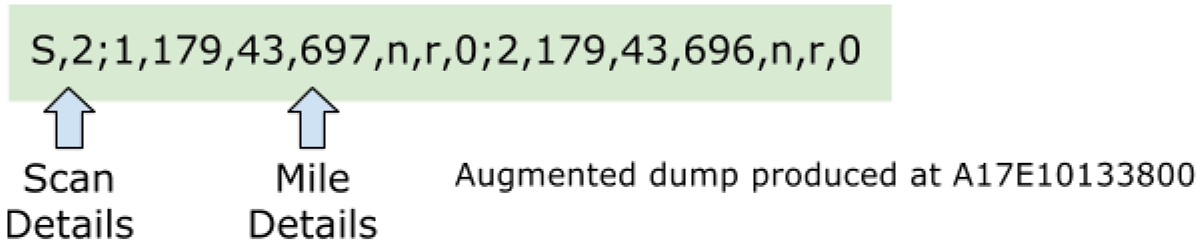
Input

This function takes a Point Zero declaration as individual params.

FindMile(x: integer, y: integer, z: integer, dir: string, itin: integer) :: string:

Output

The function produces the following result.



```
S,2;1,179,43,697,n,r,0;2,179,43,696,n,r,0
```

Augmented dump produced at A17E10133800

Scan Details

[0]	Exit condition S: SIGKILL Exit D: Duploc Exit E: EOT Exit
[1]	Miles scanned

Mile Details

[0]	Mileage
[1:4]	Track coordinates [x, y,z]
[4]	Scan direction (n/e/w/s)
[5]	Rail type P: Powered rail (any) D: Detector rail (any) S: Pressure plate (any) R: Rail (any)
[6]	Data value of rail

TrainTrax3U CodumpMgmt

Summary

TrainTrax3U CodumpMgmt - Component and Dump Manager
Author: MinecraftSBC

This module requires TrainTrax3U Core.
This module supports only Point Zero at the moment.

This module provides support for named components and fetching of dumps.

This module uses variable:

- {TrainTrax3.Param.PointZero::*}
- {TrainTrax3.Dump.PointZero::*}
- {TrainTrax3.Dump.LastUpdate}

This module provides the following functions:

- function TRTRX_GetDump(x: string) :: string:
- function TRTRX_GetRDump(x: string) :: string:
- function TRTRX_GetParam(x: string) :: string:

This module provides the following commands:

- /trx-cm < p0 > < P0 ident / - > < set / remove / list / find / dump / rdump > < ... >

Function TRTRX_GetDump / TRTRX_GetRDump

<u>Input</u>

This function takes a component identifier as string.

TRTRX_GetDump(x: string) :: string:
TRTRX_GetRDump(x: string) :: string:

<u>Output</u>

This function returns a 15-second buffered dump of the specified component and fetches new dumps *of all recorded components* if there is none available.

TRTRX_GetDump(x: string) :: string:

This function fetches real time dump of the specified component.

TRTRX_GetRDump(x: string) :: string:

For sample dumps, refer to TrainTrax3U Core: Function FindTrain / FindTrainS: Output

Function TRTRX_GetParam

Input

This function takes a component identifier as string.

function TRTRX_GetParam(x: string) :: string:

Output

This function returns the definition of the specified component.

Caution: This function is unable to return definition of Train data signs. Refer to SigNelson User Guide for more information.

For more information on component definition, refer to TrainTrax Components

Command /trx-cm

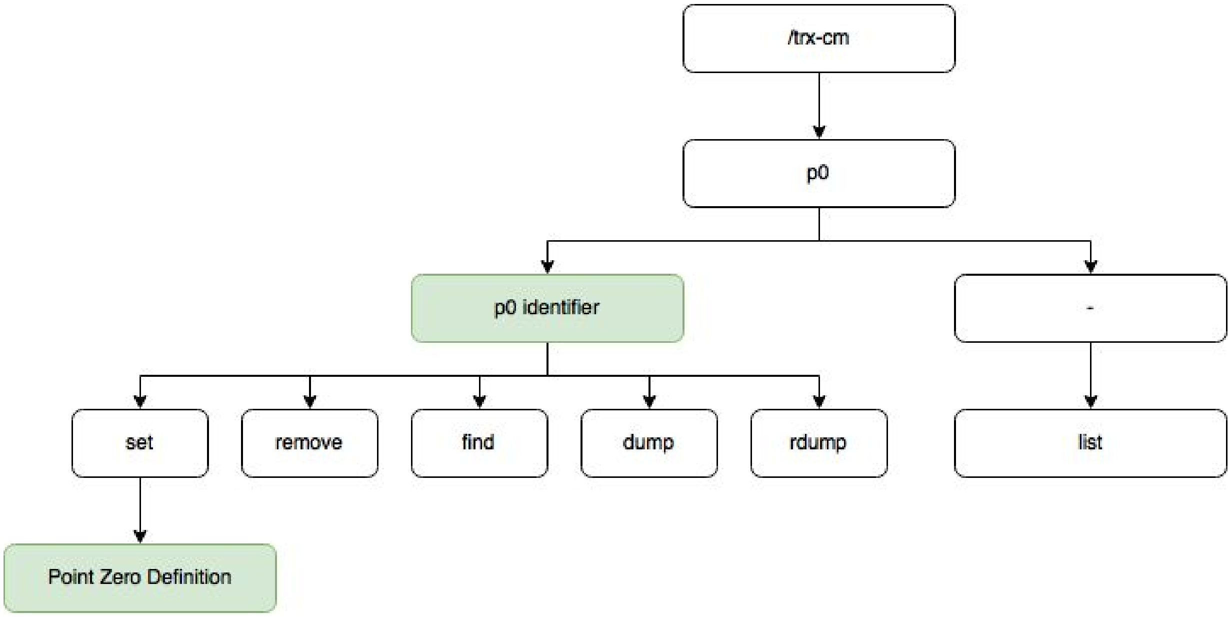
Syntax

/trx-cm <string> [<string>] [<string>] [<string>]

Aliases

/trxcm, /trtrx-cm, /trtrxcm

Content Graph



Green elements indicate evaluation elements

Description

set	Defines the component
remove	Removes the component
find	Find all specified identifier matches.
dump	Fetch the buffered dump of the component.
rdump	Fetch the real time dump of the component.
list	List out all components.

TrainTrax3U MileView

Summary

MileView - Mile Route Finder and Viewer
Author: MinecraftSBC

This module requires TrainTrax3U Core.
Component support requires TrainTrax3U CodumpMgmt.

This module provides visual and text aids regarding mileage.

This module provides the following commands:

- /trx-mv < show / find > < P0 params / @P0 ident >

Command /trx-mv

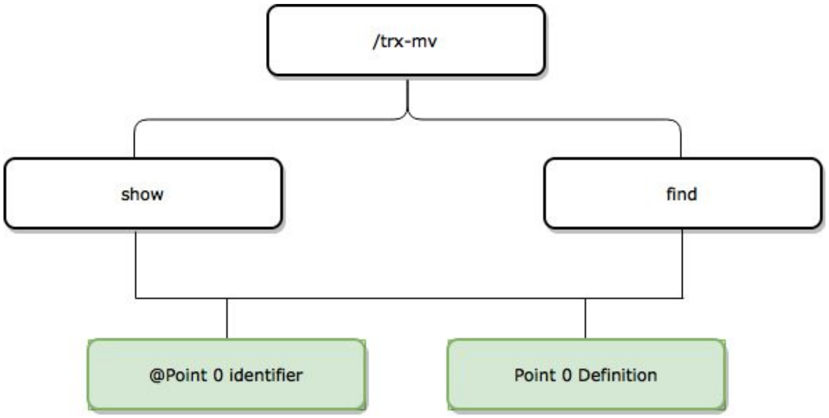
Syntax

/trx-mv <string> [<string>]

Aliases

/trxmv, /trtrx-mv, /trtrxmv

Content Graph



Description

show	Indicate the scanning path with a glowstone and text mileage description.
find	Find the closest mile to your location.

TrainTrax3U NextTrain

Summary

TrainTrax3U NextTrain - Estimated Arrival and Destination Assistant
Author: MinecraftSBC

This module requires <u>TrainTrax3U Core, TrainTrax3U CodumpMgmt and SigNelson</u>.

This module provides Next Train ETA predictions and destination evaluation, assisted SigNelson sign data completion, as well as sign evaluation functions for SigNelson.
Caution: This is the most vulnerable and head-scratching module. Be warned.

This module uses variable:

- {TrainTrax3.Player.Status::*}
- {TrainTrax3.Player.Input::*}
- {SigNelson.Sign.Line3::*}
- {SigNelson.Sign.Line4::*}

This module uses trigger:

- on rightclick on sign

This module provides the following functions:

- function TRTRX_GetNextTrain
  (dump: string, mile: number, dir: string, speed: number, dest: string) :: string:
- function TRTRX_GetNextTrainS(x: string) :: string:
- function TRTRX_NTEval(x: string) :: string:

This module provides the following commands:

- /trx-nt < rdump / view / cancel / sign / remove > < ... >

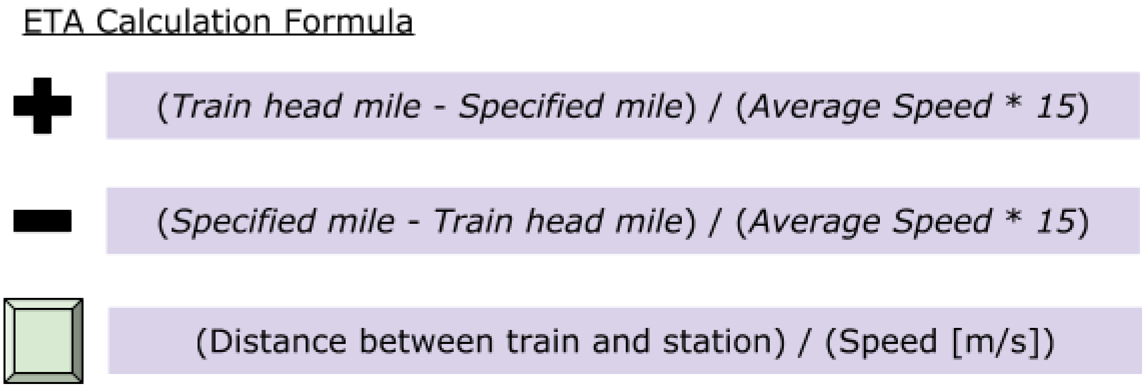
ETA Data Feed Mechanics

Summary

ETA Estimation

Dump is first fetched from TRTRX_GetDump function.

Each train has its ETA calculated with the following formula:



ETA Calculation Formula

✚	*(Train head mile - Specified mile) / (Average Speed * 15)*
━	*(Specified mile - Train head mile) / (Average Speed * 15)*
🟩	(Distance between train and station) / (Speed [m/s])

Negative ETA (Trains that have left the station) is discarded.

Trains are sorted according to their ETA by *counting sort*.

Destination Evaluation

Dump is first fetched from TRTRX_GetDump function.

Each train configuration is compared with the train config as described in destination evaluators.

The destination corresponding to the first matching train configuration is returned, with “_” symbols replaced with “ ” (space).

Sign Line Data Evaluation Mechanics

Summary

Next Train (NT)

Train ETA data is fetched from the ETA data feed.

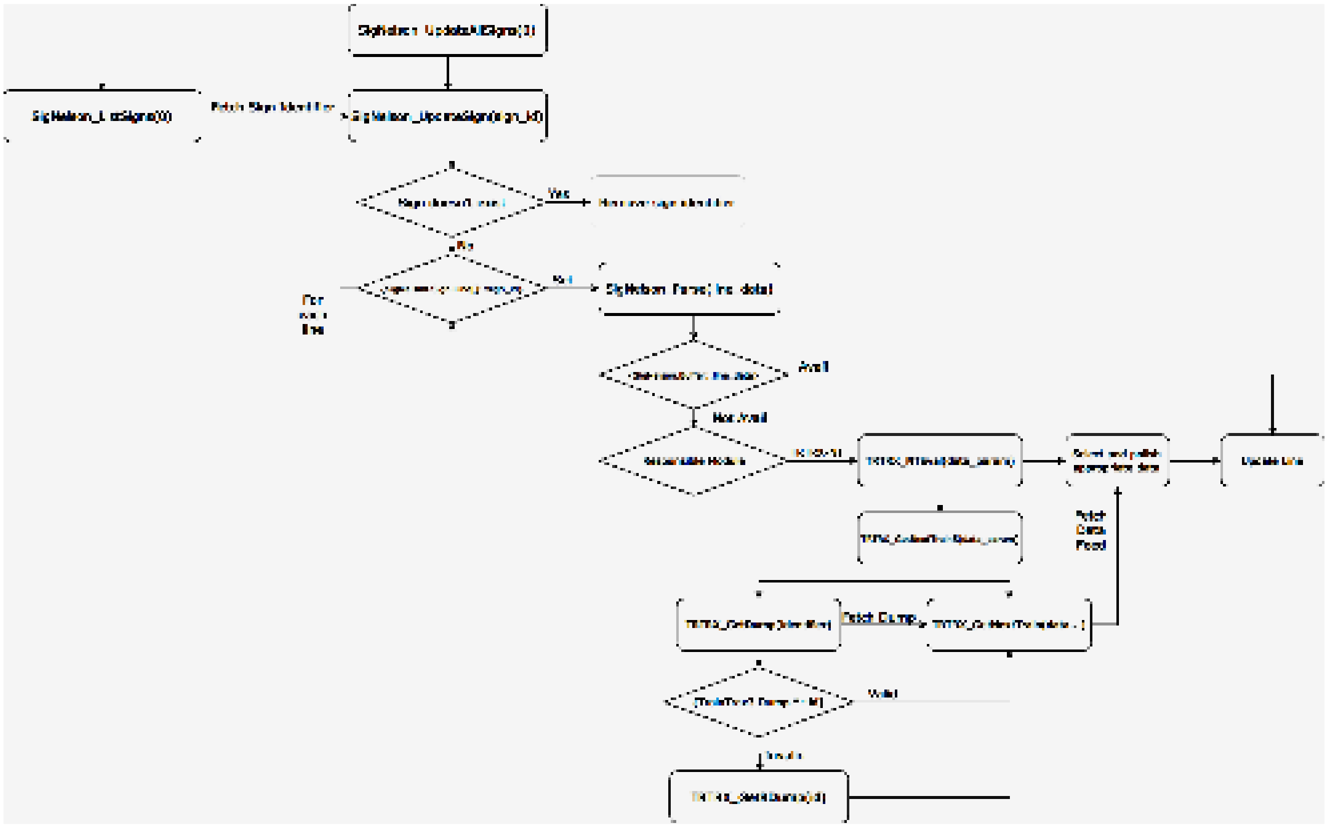
Destination and ETA is fetched accordingly.

ETA < 20 secs will be replaced with “Arr/Dep”, while others will be rounded down to the nearest integer.

The data is then polished and returned.

Sign Update Mechanics (Handled by SigNelson)

Summary Flowchart



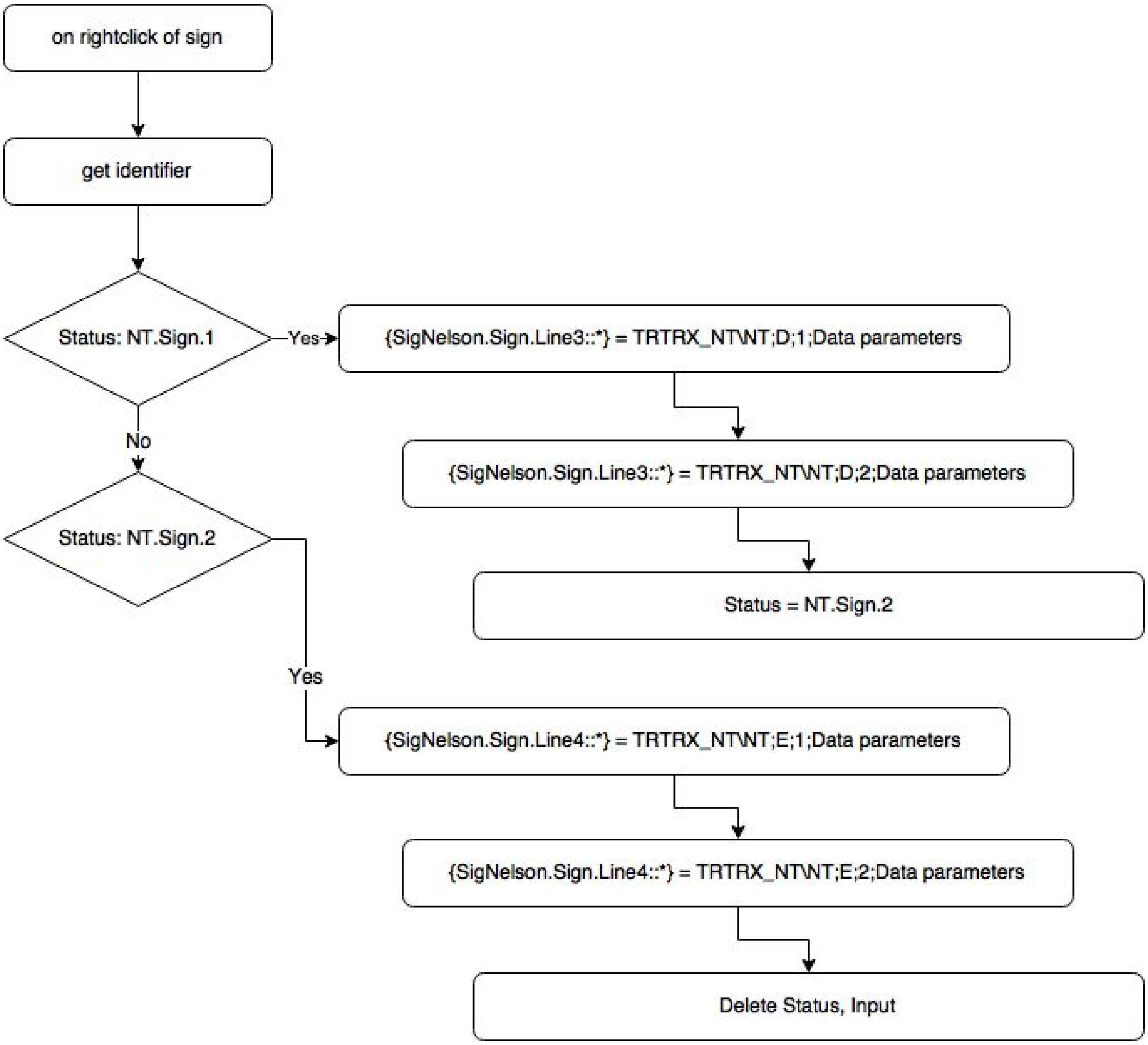
For more information on sign update procedures, refer to SigNelson’s User Guide.

Usage of Trigger: on rightclick on sign

Summary

This trigger is used for the assisted setup of ETA sign data for SigNelson.

Flowchart


on rightclick of sign
get identifier
Status: NT.Sign.1
Yes
{SigNelson.Sign.Line3::*} = TRTRX_NT\NT;D;1;Data parameters
No
{SigNelson.Sign.Line3::*} = TRTRX_NT\NT;D;2;Data parameters
Status: NT.Sign.2
Status = NT.Sign.2
Yes
{SigNelson.Sign.Line4::*} = TRTRX_NT\NT;E;1;Data parameters
{SigNelson.Sign.Line4::*} = TRTRX_NT\NT;E;2;Data parameters
Delete Status, Input

Function TRTRX_GetNextTrain / TRTRX_GetNextTrainS

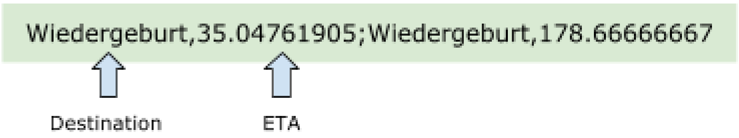
Input

This function takes a data parameter as individual elements.
function TRTRX_GetNextTrain
(dump: string, mile: number, dir: string, speed: number, dest: string) :: string:

This function takes a data parameter as a continuous string.
function TRTRX_GetNextTrainS(x: string) :: string:

Output

The function produces the following result -



```
Wiedergeburt,35.04761905;Wiedergeburt,178.66666667
```

Sample dump from data parameter [p0 nsl_up 1909 - 0.7 m:Wiedergeburt] at A17E10235131

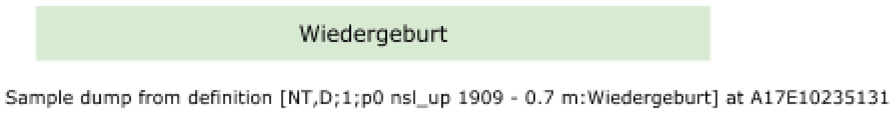
Function TRTRX_NTEval

Input

This function takes a Next Train sign definition as a continuous string.
function TRTRX_NTEval(x: string) :: string:

Output

The function produces a single string of the evaluated line data.



Wiedergeburt

Sample dump from definition [NT,D;1;p0 nsl_up 1909 - 0.7 m:Wiedergeburt] at A17E10235131

Command /trx-nt

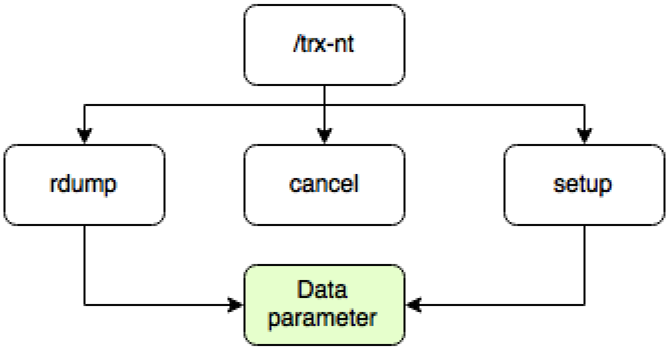
Syntax

/trx-cm <string>

Aliases

/trxnt, /trtrx-nt, /trtrxnt

Content Graph



Green elements indicate evaluation elements

Description

rdump	Fetches the real time dump of a data parameter.
cancel	Cancels the operation and reset all running variables.
setup	Starts the setup wizard for ETA signs.

For more information on data parameters, refer to
TrainTrax Components: Train data signs (Handled by SigNelson): Declaration: Next Train Information