EQUIPMENT DEFECT DETECTOR SUPPORT MANUAL

MICRO HOT BOX DETECTOR/TALKER

These instructions do not purport to cover all details or variations in equipment represented in-game nor to provide for every possible contingency to be met in connection with operation. Should further information be desired or should particular problems arise which are not covered sufficiently for the user’s purposes, the matter should be referred to Searchlight Simulations LLC.

EULA

SOFTWARE LICENSE AGREEMENT

1. SOFTWARE LICENSE AGREEMENT

This end user license agreement is a legal contract between you and Searchlight Simulations.

The following terms have the following meanings:

"License" means the license to use the Product as set out in paragraph 3.

"Searchlight Simulations" means Searchlight Simulations, a company incorporated in the United States of America.

"Product" means the "Equipment Defect Detector" and the with it associated "LUA Source Code".

By proceeding to download or install the Product, and in consideration of your use of the Product, you are deemed to agree to be bound by the terms of this License.

Searchlight Simulations permits you to use the Product only in accordance with the terms of this License and your rights under this License will terminate automatically without notice if you fail to comply with the terms of this License. Should your rights under this License terminate for any reason you must immediately cease all further use of the Product and destroy all copies of it.

ANY USE OF THIS SOFTWARE IN A COMMERCIAL ENVIRONMENT, AND ANY RESALE OR FURTHER DISTRIBUTION OF THE SOFTWARE, OTHER THAN AS EXPRESSLY AUTHORIZED BY THIS AGREEMENT, CONSTITUTES A BREACH OF THIS AGREEMENT AND MAY VIOLATE APPLICABLE COPYRIGHT LAWS.

2. WARRANTY DISCLAIMER

THIS PRODUCT IS DELIVERED TO YOU 'AS IS' AND WITH ALL FAULTS. SEARCHLIGHT SIMULATIONS DO NOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THIS PRODUCT. EXCEPT TO THE EXTENT THAT ANY WARRANTY, CONDITION, REPRESENTATION OR TERM CANNOT BE EXCLUDED OR LIMITED BY LAW, WE MAKE NO WARRANTIES, CONDITIONS, REPRESENTATIONS OR OTHER TERMS (EXPRESS OR IMPLIED WHETHER BY STATUTE, COMMON LAW, CUSTOM, USAGE OR OTHERWISE) AS TO ANY MATTER INCLUDING, WITHOUT LIMITATION, NON-INFRINGEMENT OF THIRD PARTY RIGHTS, SATISFACTORY QUALITY AND FITNESS FOR PURPOSE. THE PROVISIONS OF THIS SECTION 2 SHALL SURVIVE THE TERMINATION OF THIS LICENSE HOWSOEVER CAUSED.

3. PRODUCT LICENSE

In consideration of you agreeing to abide by the terms of this License and subject to your compliance with the terms of this License, Searchlight Simulations grants you a non-exclusive, non-transferable license to use the Product for the following purposes and in the following manner:

> You may NOT resell, charge for, sub-license, rent, lease, loan or otherwise commercially exploit the Product without our prior written consent. We reserve the right to withdraw any such consent (or part thereof) for any reason and without notice and to demand that you immediately cease any activity in respect of which permission is withdrawn.
>
> You may translate, adapt, vary, modify, alter, create derivative works based upon, or integrate any other computer programs with, the Product in whole or in part.
>
> You may repackage or redistribute the Product in whole or in part for noncommercial applications ONLY and ONLY if providing a copy, link or other form of this end user license agreement bundled with your application.
>
> You may NOT use the Product to engage in or allow others to engage in any illegal activity.
>
> You may NOT transfer or assign your rights or obligations under this License to any person.
>
> You are NOT permitted to grant any sub-licenses of the Product.
>
> You may NOT use the Product in corporate or commercial environments.
>
> You may NOT use the Product to sell or deliver any other product,service or application to a third party or other commercial business.

EULA

4. INTELLECTUAL PROPERTY RIGHTS

The Product contains intellectual property of Searchlight Simulations and is protected by law. You acknowledge that all intellectual property rights in the Product anywhere in the world belong to Searchlight Simulations, that rights in the Product are licensed (not sold) to you, and that you have no rights in, or to, the Product other than the right to use it in accordance with the terms of this License.

5. LIMITATION OF LIABILITY

IN NO CIRCUMSTANCES WILL Searchlight Simulations BE LIABLE TO YOU FOR ANY DAMAGES, CLAIMS OR COSTS WHATSOEVER (WHETHER IN CONTRACT, TORT, NEGLIGENCE OR OTHERWISE) INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, LOST SAVINGS, LOST INFORMATION, LOSS OF DATA NOR ANY CONSEQUENTIAL, INDIRECT, INCIDENTAL DAMAGES. THE FOREGOING LIMITATIONS AND EXCLUSION APPLY TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION.

Nothing in this License limits Searchlight Simulations' liability to you in the event of death or personal injury resulting from our negligence or for fraud. The provisions of this section 5 shall survive the termination of this License howsoever caused.

6. LAW AND JURISDICTION

This License and any dispute or claim arising out of or in connection with it or its subject matter or its formation (including non-contractual disputes or claims) shall be governed by and construed in accordance with US law and submitted to the non-exclusive jurisdiction of the US courts.

7. GENERAL

If any part of this License is found to be void and unenforceable it will not affect the validity of the remainder of this License, which shall remain valid and enforceable according to its terms. This License may only be varied in writing and signed by an authorized representative of Searchlight Simulations. However, notwithstanding the foregoing, Searchlight Simulations may revise this agreement from time to time by posting the most current version on its website. If a revision meaningfully reduces your rights, Searchlight Simulations will notify you (by, for example, sending a message to the email address associated with your account, posting on its facebook page or on its webpage). By continuing to use the Product after the revisions come into effect, you agree to be bound by the revised agreement. Updates may be licensed to you by Searchlight Simulations with additional or different terms but Searchlight Simulations has no obligation to provide any updates. This License is the entire agreement between you and us and supersedes any prior representations, undertakings or advertising relating to the Product and you acknowledge that in entering into this License you have not relied on any statement, representation, advertising, assurance or warranty (whether made negligently or innocently) other than as expressly set out in this License. Except as expressly set forth in this agreement, the exercise by either party of any of its remedies under this agreement will be without prejudice to its other remedies under this agreement or otherwise. All notices or approvals required or permitted under this agreement will be in writing and delivered by email (if available), and in each instance will be deemed given upon receipt. The failure by either party to enforce any provision of this agreement will not constitute a waiver of future enforcement of that or any other provision. Searchlight Simulations is entitled to transfer or assign any of its rights or obligations under this agreement. You may not assign any of your rights under this agreement, and any such attempt will be void.

[07292020]

GENERAL AND INSTALLATION

GENERAL

Our equipment defect detectors are (unlike anything ever done before) fully functional and do not require any custom scenario scripting or else to work. Once your locomotive and rolling stock is properly configured, all you need to do is place a detector down (through the world editor), assign a valid ID to it and it's ready. The detectors can report actual axle count, car count, train length, speed and ambient air temperature. The user is able to choose what specific railroad the detector should be configured to, what specific milepost it's located at and whether or not the detector is supposed to report train length, car count, speed or temperature respectfully. Axle count and milepost location are always reported, no matter the railroad or user input. Between certain railroads you'll be able to observe slight differences in the detector report, all specific to the prototype (were applicable).

The equipment defect detector is capable of reporting axle count up to 9999, car count up to 9999, train length up to 99999FT, train speed up to 999MPH and ambient air temperature from up to 99F°.

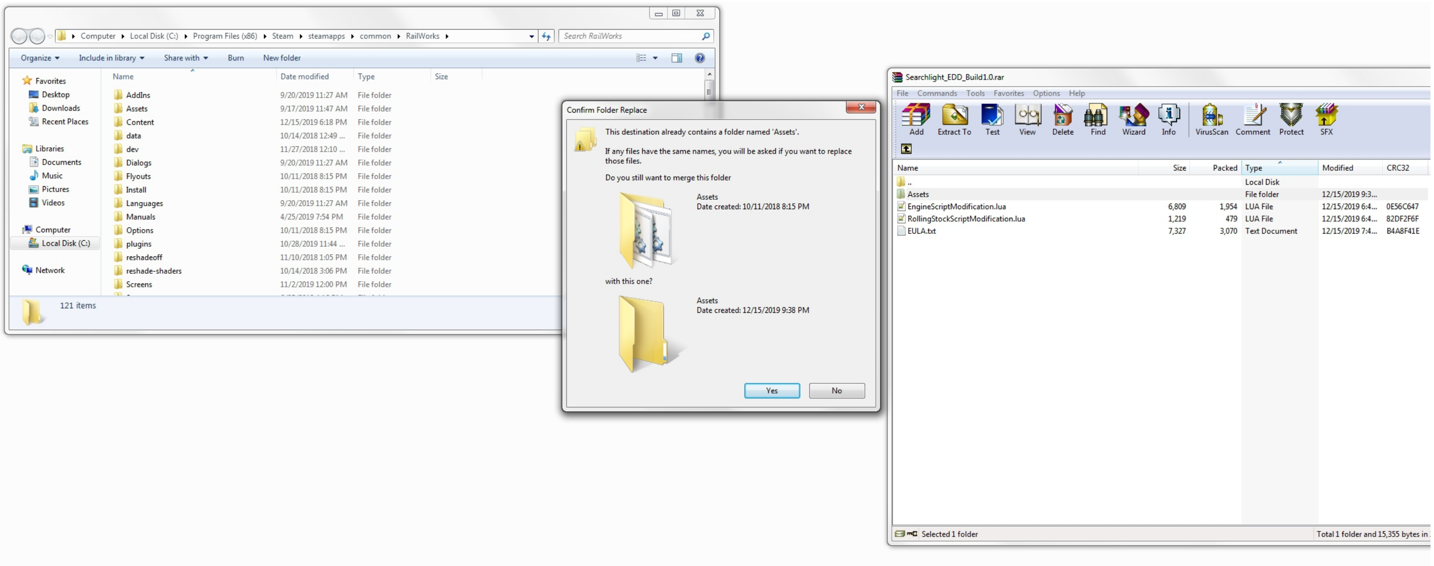
INSTALLATION

> *NOTE: BECAUSE THE EQUIPMENT DEFECT DETECTORS REQUIRE MANUAL MOVING/EDITING OF FILES REGARDLESS, WE WON'T PROVIDE A SELF-INSTALLER FOR IT. ALL REQUIRED FILES CAN BE FOUND IN THE INCLUDED ARCHIVE (.RAR FILE EXTENSION).*

In order to access the files inside the archive you will require a free or commercial copy of either WinRAR or 7ZIP, available from the links below.

– *www.7-zip.org*
– *www.win-rar.com*

Inside the archive you'll find a folder labeled “Assets”, two separate LUA source files and our EULA. To install the actual equipment defect detector, open up your “Railworks” directory and drag the “Assets” folder into it. When asked to overwrite/replace the existing folder, make sure to click yes/allow it.



Name	Date modified	Type	Size
AddIns	9/20/2019 11:27 AM	File folder	
Assets	9/17/2019 11:47 AM	File folder	
Content	12/15/2019 6:18 PM	File folder	
data	10/14/2018 12:49 ...	File folder	
dev	11/27/2018 12:10 ...	File folder	
Dialogs	9/20/2019 11:27 AM	File folder	
Flyouts	10/11/2018 8:15 PM	File folder	
Install	10/11/2018 8:15 PM	File folder	
Languages	9/20/2019 11:27 AM	File folder	
Manuals	4/25/2019 7:54 PM	File folder	
Options	10/11/2018 8:15 PM	File folder	
plugins	10/28/2019 11:44 ...	File folder	
reshadeoff	11/10/2018 1:05 PM	File folder	
reshade-shaders	10/14/2018 3:06 PM	File folder	
Screens	11/2/2019 12:00 PM	File folder	

121 items

Confirm Folder Replace

This destination already contains a folder named 'Assets'.

If any files have the same names, you will be asked if you want to replace those files.

Do you still want to merge this folder

Assets
Date created: 10/11/2018 8:15 PM

with this one?

Assets
Date created: 12/15/2019 9:38 PM

Name	Size	Packed	Type	Modified	CRC32
..			Local Disk		
Assets			File folder	12/15/2019 9:3...	
EngineScriptModification.lua	6,809	1,954	LUA File	12/15/2019 6:4...	0E56C647
RollingStockScriptModification.lua	1,219	479	LUA File	12/15/2019 6:4...	82DF2F6F
EULA.txt	7,327	3,070	Text Document	12/15/2019 7:4...	B4A8F41E

Selected 1 folder
Total 1 folder and 15,355 bytes in

PREPERATION

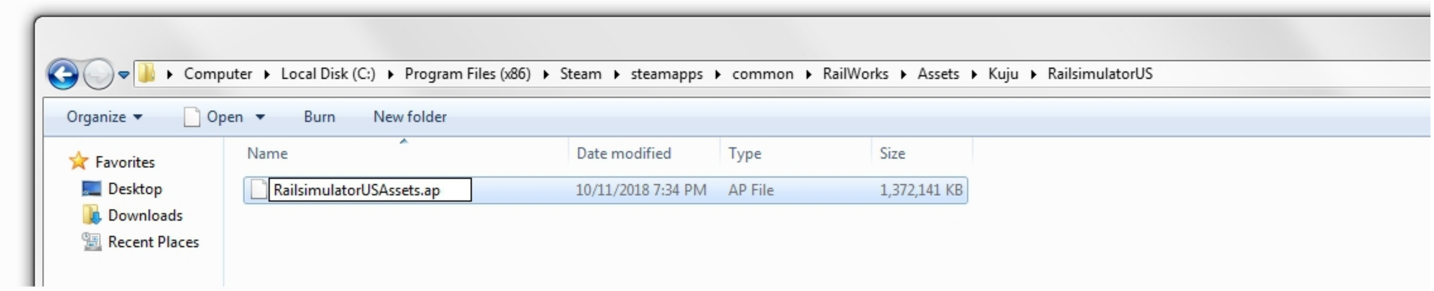
BEFORE GETTING STARTED

NOTE: THE FOLLOWING STEPS NEED TO BE PERFORMED ON EVERY LOCOMOTIVE/DLC INDIVIDUALLY (PROVIDED THE LOCOMOTIVE IN QUESTION IS RUNNING ON IT'S OWN, UNIQUE ENGINE SCRIPT). TO DETERMINE WHAT ENGINE SCRIPT THE LOCOMOTIVE YOU'RE INTENDING TO EQUIP WITH OUR EQUIPMENT DEFECT DETECTOR CAPABILITY IS CALLING FOR (UNLESS YOU ALREADY KNOW IT), YOU NEED TO PERFORM THE FOLLOWING:

First you have to find the locomotive's binary (.bin) file. For the example below we'll be using the “Black” Kuju SD40-2.

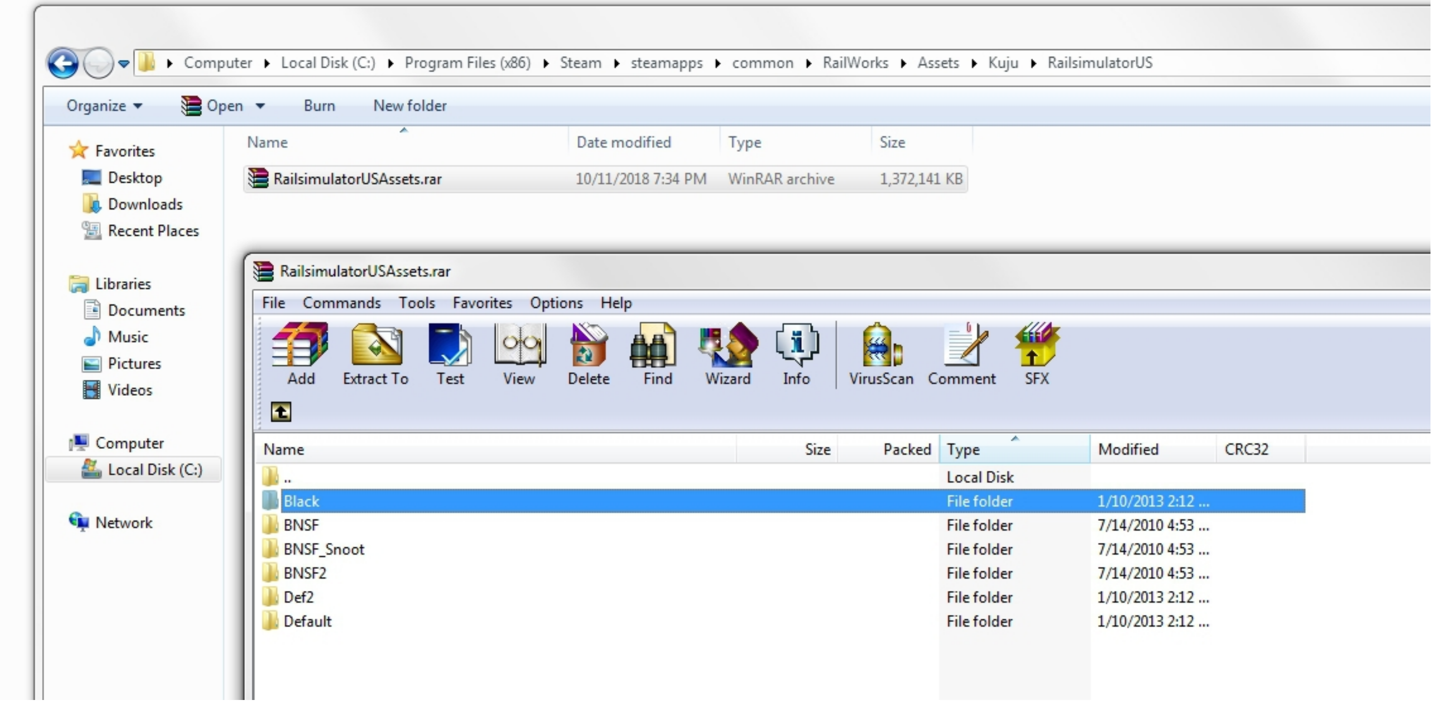
NOTE: IF YOU STUMBLE ACROSS A PACKAGE FILE (.AP EXTENSION), CHANGE IT'S EXTENSION FROM .AP TO EITHER .RAR (WINRAR) OR .ZIP (7ZIP) IN ORDER TO EASILY ACCESS IT'S CONTENTS. THE SAME CAN AUTOMATICALLY BE DONE THROUGH RW-TOOLS.

NOTICE: Should you already have everything unpacked and ready to go, skip to page 4.



Computer ▸ Local Disk (C:) ▸ Program Files (x86) ▸ Steam ▸ steamapps ▸ common ▸ RailWorks ▸ Assets ▸ Kuju ▸ RailsimulatorUS

Name	Date modified	Type	Size
RailsimulatorUSAssets.ap	10/11/2018 7:34 PM	AP File	1,372,141 KB



Computer ▸ Local Disk (C:) ▸ Program Files (x86) ▸ Steam ▸ steamapps ▸ common ▸ RailWorks ▸ Assets ▸ Kuju ▸ RailsimulatorUS

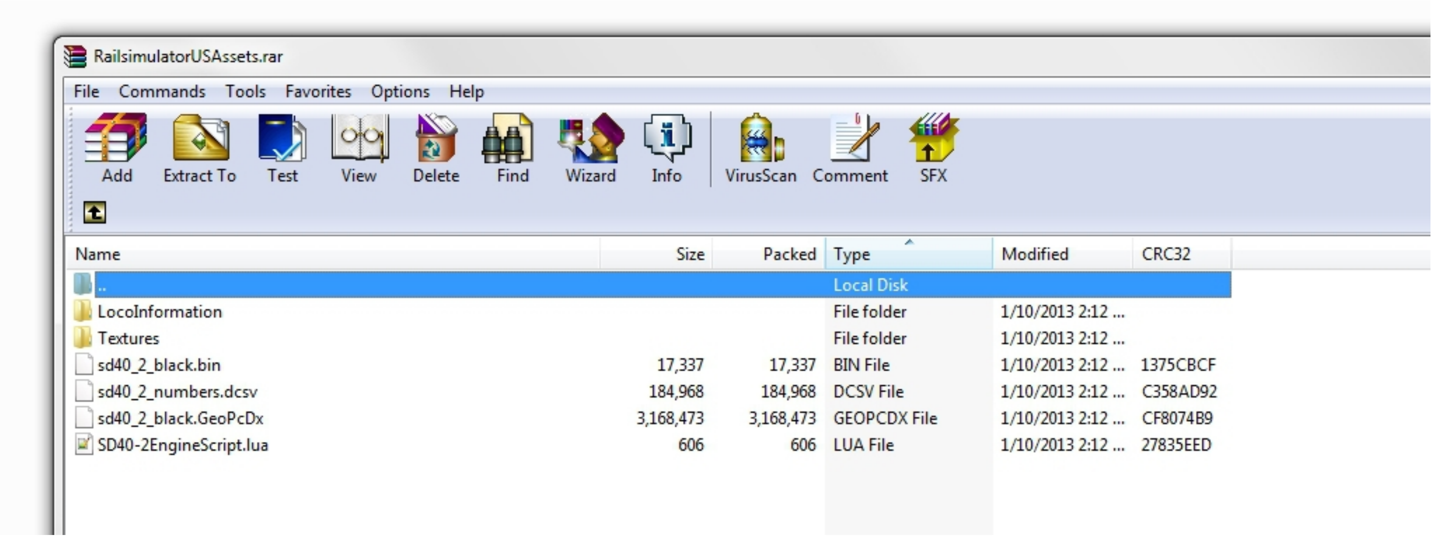
Name	Date modified	Type	Size
RailsimulatorUSAssets.rar	10/11/2018 7:34 PM	WinRAR archive	1,372,141 KB

RailsimulatorUSAssets.rar

Name	Size	Packed	Type	Modified	CRC32
..			Local Disk		
Black			File folder	1/10/2013 2:12 ...	
BNSF			File folder	7/14/2010 4:53 ...	
BNSF_Snoot			File folder	7/14/2010 4:53 ...	
BNSF2			File folder	7/14/2010 4:53 ...	
Def2			File folder	1/10/2013 2:12 ...	
Default			File folder	1/10/2013 2:12 ...	

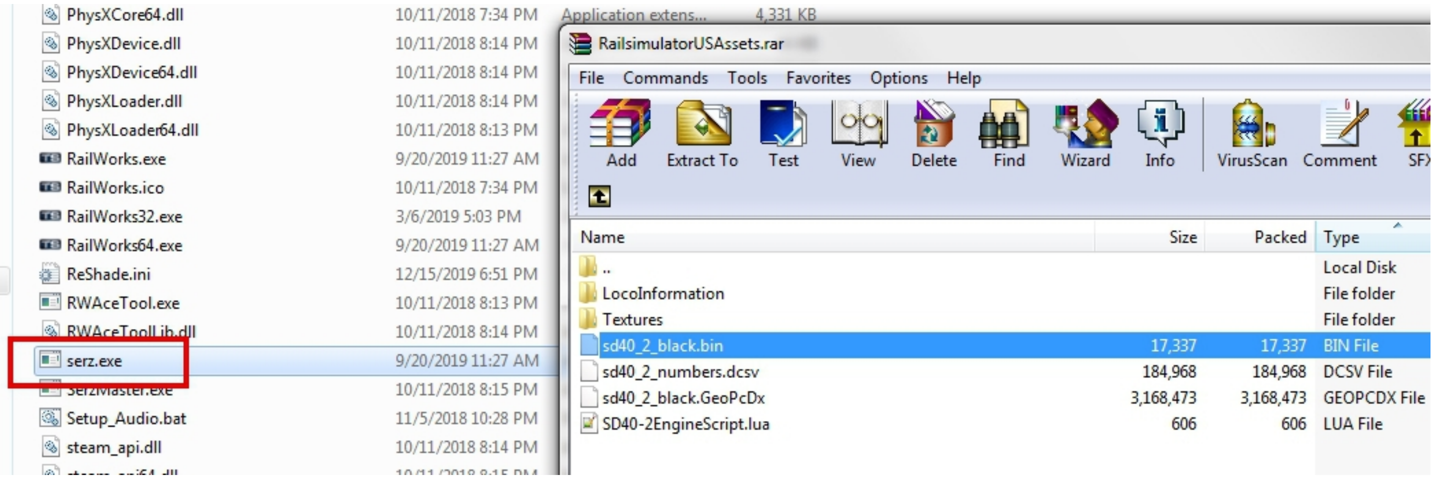
PREPERATION

Browse through the “RailVehicles” folder until you find your desired locomotive. In our example we can instantly spot an engine script file when opening the “Black” engine sub-folder. This tells us (without having to even convert the “sd40_2_black.bin” file, that the engine script called by the “sd40_2_black.bin” is the “SD40-2EngineScript.lua” file found in the same location.


RailsimulatorUSAssets.rar
File Commands Tools Favorites Options Help
Add
Extract To
Test
View
Delete
Find
Wizard
Info
VirusScan
Comment
SFX

Name	Size	Packed	Type	Modified	CRC32
..			Local Disk		
LocoInformation			File folder	1/10/2013 2:12 ...	
Textures			File folder	1/10/2013 2:12 ...	
sd40_2_black.bin	17,337	17,337	BIN File	1/10/2013 2:12 ...	1375CBCF
sd40_2_numbers.dcsv	184,968	184,968	DCSV File	1/10/2013 2:12 ...	C358AD92
sd40_2_black.GeoPcDx	3,168,473	3,168,473	GEOPCDX File	1/10/2013 2:12 ...	CF8074B9
SD40-2EngineScript.lua	606	606	LUA File	1/10/2013 2:12 ...	27835EED

NOTE: IF YOU DON'T HAPPEN TO SPOT AN ENGINE SCRIPT FILE (EITHER .LUA OR .OUT EXTENSION), YOU WILL HAVE TO DRAG THE ENGINE .BIN FILE INTO THE “SERZ.EXE” LOCATED IN YOUR MAIN RAILWORKS DIRECTORY IN ORDER TO CONVERT IT INTO AN EXTENSIBLE MARKUP LANGUAGE DATA FILE (.XML EXTENSION) WHICH YOU CAN THEN OPEN IN A TEXT EDITOR LIKE NOTEPAD++. THIS HOWEVER WILL ONLY WORK IF YOU FIRST DRAG/MOVE THE ENGINE .BIN FILE INTO A TEMPORARY LOCATION, OUTSIDE THE CURRENTLY OPEN WINRAR OR 7ZIP BROWSER.

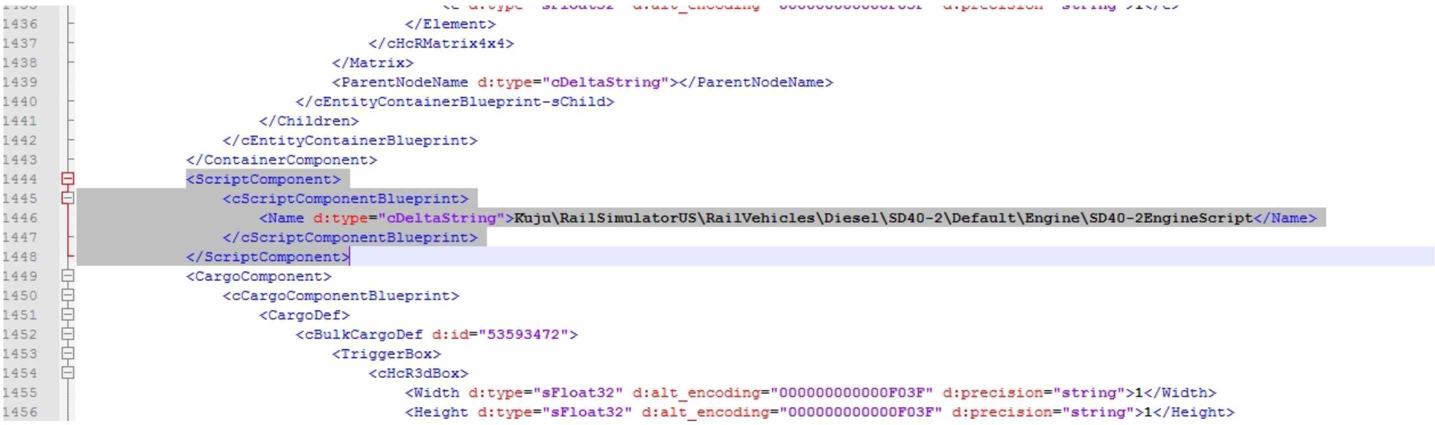

RailsimulatorUSAssets.rar
File Commands Tools Favorites Options Help
Add
Extract To
Test
View
Delete
Find
Wizard
Info
VirusScan
Comment
SFX

PhysXCore64.dll	10/11/2018 7:34 PM	Application extens...	4,331 KB
PhysXDevice.dll	10/11/2018 8:14 PM		
PhysXDevice64.dll	10/11/2018 8:14 PM		
PhysXLoader.dll	10/11/2018 8:14 PM		
PhysXLoader64.dll	10/11/2018 8:13 PM		
RailWorks.exe	9/20/2019 11:27 AM		
RailWorks.ico	10/11/2018 7:34 PM		
RailWorks32.exe	3/6/2019 5:03 PM		
RailWorks64.exe	9/20/2019 11:27 AM		
ReShade.ini	12/15/2019 6:51 PM		
RWAceTool.exe	10/11/2018 8:13 PM		
RWAceToolLib.dll	10/11/2018 8:14 PM		
serz.exe	9/20/2019 11:27 AM		
SerzMaster.exe	10/11/2018 8:15 PM		
Setup_Audio.bat	11/5/2018 10:28 PM		
steam_api.dll	10/11/2018 8:14 PM		
steam_api64.dll	10/11/2018 8:15 PM		

Name	Size	Packed	Type
..			Local Disk
LocoInformation			File folder
Textures			File folder
sd40_2_black.bin	17,337	17,337	BIN File
sd40_2_numbers.dcsv	184,968	184,968	DCSV File
sd40_2_black.GeoPcDx	3,168,473	3,168,473	GEOPCDX File
SD40-2EngineScript.lua	606	606	LUA File

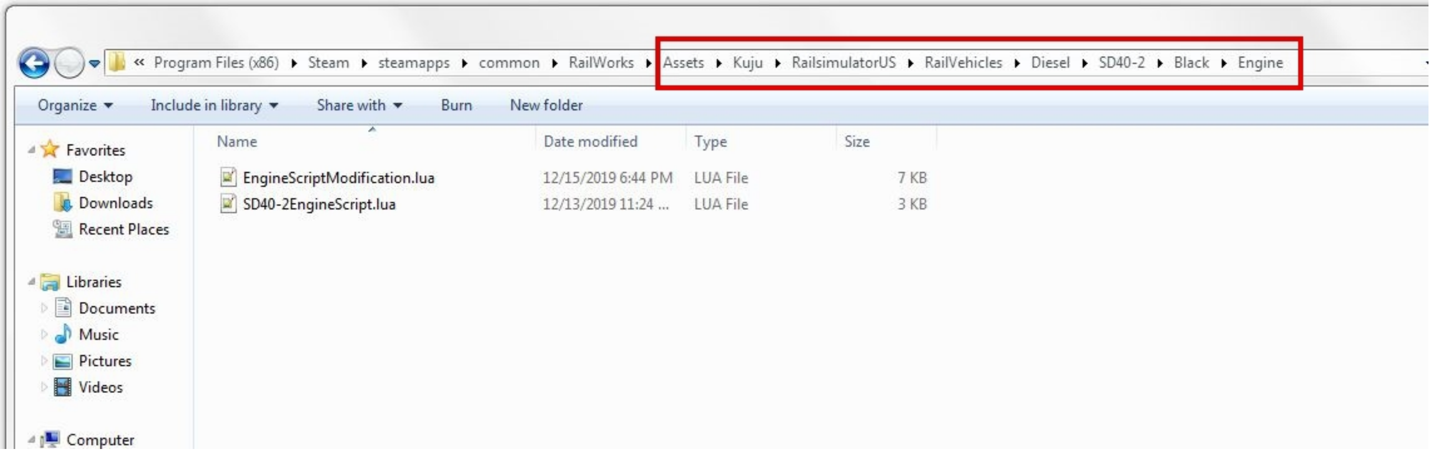
PREPERATION

If done properly, the “serz.exe” will create an .xml version of the original .bin file. You then have to open that .xml file (we're using Notepad++ in our example) and scroll all the way to the bottom. There you'll find the full path along with the actual engine script name the locomotive is calling for. That's the engine script we'll be modifying in the next step.



```
                        </Element>
                    </cHcRMatrix4x4>
                </Matrix>
                <ParentNodeName d:type="cDeltaString"></ParentNodeName>
            </cEntityContainerBlueprint-sChild>
        </Children>
    </cEntityContainerBlueprint>
</ContainerComponent>
<ScriptComponent>
    <cScriptComponentBlueprint>
        <Name d:type="cDeltaString">Kuju\RailSimulatorUS\RailVehicles\Diesel\SD40-2\Default\Engine\SD40-2EngineScript</Name>
    </cScriptComponentBlueprint>
</ScriptComponent>
<CargoComponent>
    <cCargoComponentBlueprint>
        <CargoDef>
            <cBulkCargoDef d:id="53593472">
                <TriggerBox>
                    <cHcR3dBox>
                        <Width d:type="sFloat32" d:alt_encoding="000000000000F03F" d:precision="string">1</Width>
                        <Height d:type="sFloat32" d:alt_encoding="000000000000F03F" d:precision="string">1</Height>
```

If not already present, make sure to create a full folder path as indicated in “ScriptComponentBlueprint” section and copy the referenced engine script file into said location.



« Program Files (x86) ▸ Steam ▸ steamapps ▸ common ▸ RailWorks ▸ Assets ▸ Kuju ▸ RailsimulatorUS ▸ RailVehicles ▸ Diesel ▸ SD40-2 ▸ Black ▸ Engine

Name	Date modified	Type	Size
EngineScriptModification.lua	12/15/2019 6:44 PM	LUA File	7 KB
SD40-2EngineScript.lua	12/13/2019 11:24 ...	LUA File	3 KB

Next copy or drag our “EngineScriptModification.lua” file into the same folder. There's two things you have to do now. First you have to change the file extension of the original locomotive script file “SD40-2EngineScript.lua” (in our example) to “SD40-2EngineScript.lua.prev”. For an .out file “SD40-2EngineScript.out” (as an example), you would change it to “SD40-2EngineScript.out.prev”. Simple enough, right?

Once this is done, rename the “EngineScriptModification.lua” into what the original locomotive script file used to be. So in our example we'll rename the “EngineScriptModification.lua” file into “SD40-2EngineScript.lua”.

> *NOTE: It is very important to keep the file name of our replacement script file the exact same as the original, otherwise the game will disregard it and thus any changes/modifications made to it.*

LOCOMOTIVE SCRIPT MODIFICATION

LOCOMOTIVE SCRIPT MODIFICATION

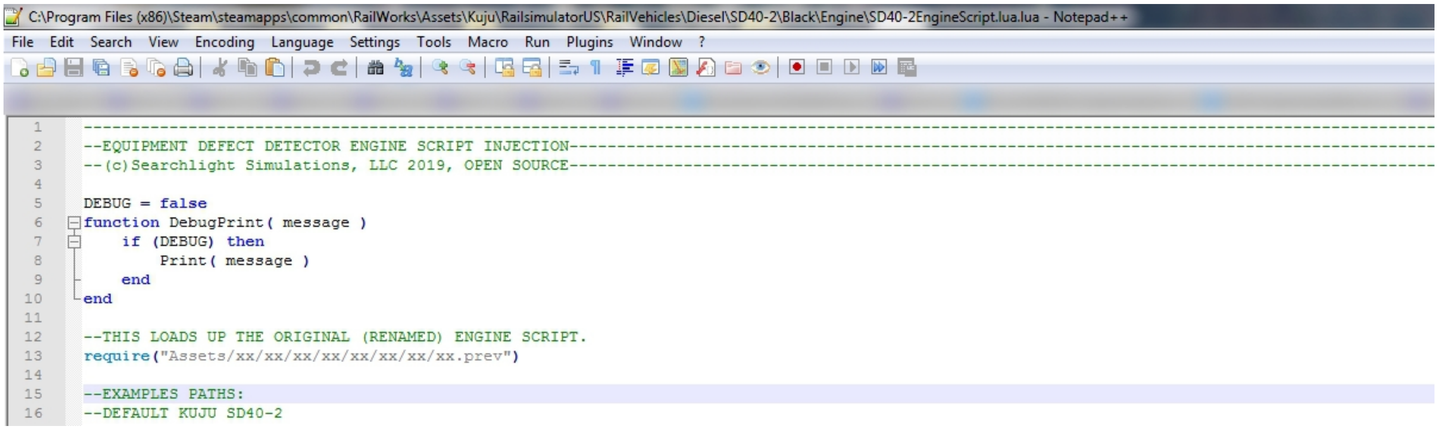
The following steps will enable your locomotive and rolling stock to “talk” with our equipment defect detector and are essential for it's operation.

For the locomotive script, there's three things you'll have to do. First of all, you have to open the new, replacement engine script “SD40-2EngineScript.lua” (in our example) in a text editor like Notepad++ and replace line 13 (from the top) “require("Assets/xx/xx/xx/xx/xx/xx/xx/xx.prev” with the actual full path and name of the original (now renamed) locomotive script file.

> *NOTE: MAKE SURE TO KEEP THE FORMATTING EXACTLY AS IS. BACKSLASHES IN THE FILE PATH WILL NOT WORK.*

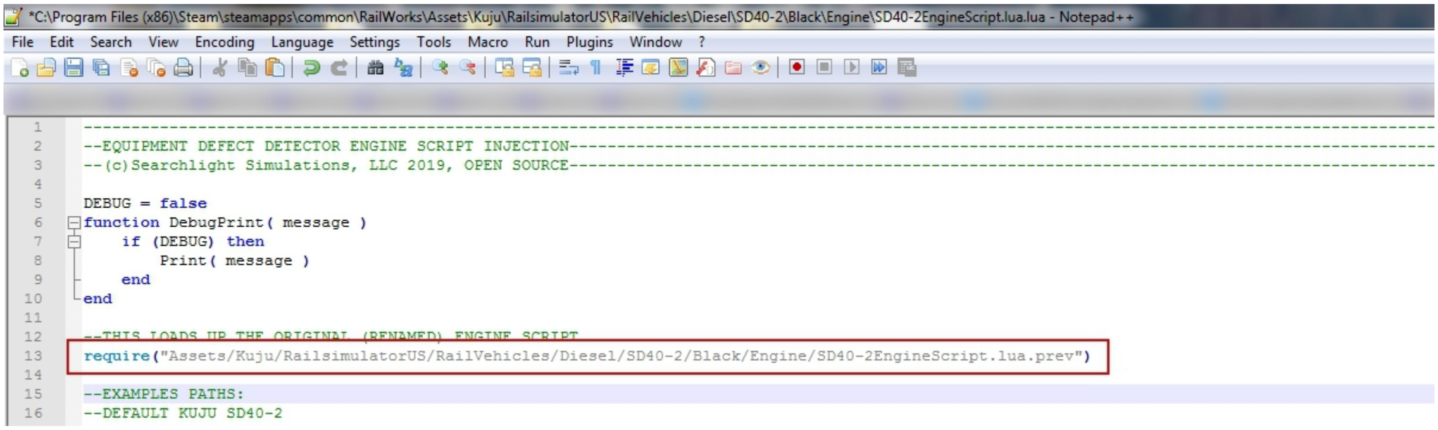
For our example, we'll change the entire path to look like the following:

BEFORE


C:\Program Files (x86)\Steam\steamapps\common\RailWorks\Assets\Kuju\RailsimulatorUS\RailVehicles\Diesel\SD40-2\Black\Engine\SD40-2EngineScript.lua.lua - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

```
1   ------------------------------------------------------------------------------------------------------------------
2   --EQUIPMENT DEFECT DETECTOR ENGINE SCRIPT INJECTION---------------------------------------------------------------
3   --(c)Searchlight Simulations, LLC 2019, OPEN SOURCE---------------------------------------------------------------
4
5   DEBUG = false
6   function DebugPrint( message )
7       if (DEBUG) then
8           Print( message )
9       end
10  end
11
12  --THIS LOADS UP THE ORIGINAL (RENAMED) ENGINE SCRIPT.
13  require("Assets/xx/xx/xx/xx/xx/xx/xx/xx.prev")
14
15  --EXAMPLES PATHS:
16  --DEFAULT KUJU SD40-2
```

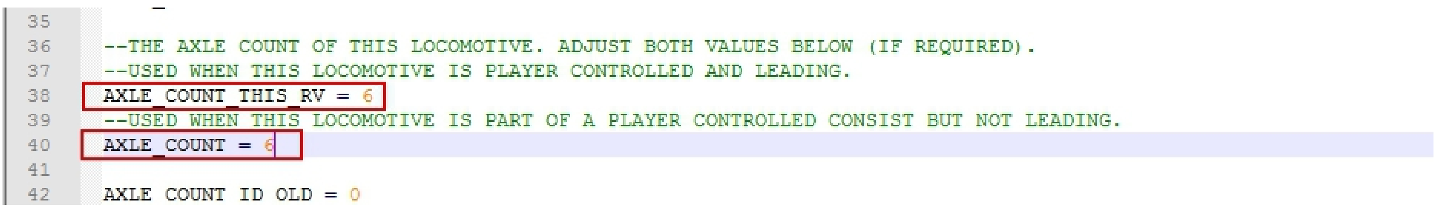
AFTER


*C:\Program Files (x86)\Steam\steamapps\common\RailWorks\Assets\Kuju\RailsimulatorUS\RailVehicles\Diesel\SD40-2\Black\Engine\SD40-2EngineScript.lua.lua - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

```
1   ------------------------------------------------------------------------------------------------------------------
2   --EQUIPMENT DEFECT DETECTOR ENGINE SCRIPT INJECTION---------------------------------------------------------------
3   --(c)Searchlight Simulations, LLC 2019, OPEN SOURCE---------------------------------------------------------------
4
5   DEBUG = false
6   function DebugPrint( message )
7       if (DEBUG) then
8           Print( message )
9       end
10  end
11
12  --THIS LOADS UP THE ORIGINAL (RENAMED) ENGINE SCRIPT.
13  require("Assets/Kuju/RailsimulatorUS/RailVehicles/Diesel/SD40-2/Black/Engine/SD40-2EngineScript.lua.prev")
14
15  --EXAMPLES PATHS:
16  --DEFAULT KUJU SD40-2
```

LOCOMOTIVE SCRIPT MODIFICATION

Next you want to make sure the axle count of your locomotive matches the assigned variables in line 38 and line 40.
For a six-axle locomotive, both variables should read “6”, for a four-axle locomotive, you have to change the variables to each read “4” etc..



```
35
36    --THE AXLE COUNT OF THIS LOCOMOTIVE. ADJUST BOTH VALUES BELOW (IF REQUIRED).
37    --USED WHEN THIS LOCOMOTIVE IS PLAYER CONTROLLED AND LEADING.
38    AXLE_COUNT_THIS_RV = 6
39    --USED WHEN THIS LOCOMOTIVE IS PART OF A PLAYER CONTROLLED CONSIST BUT NOT LEADING.
40    AXLE_COUNT = 6
41
42    AXLE COUNT ID OLD = 0
```

Once done, save any changes made and close the replacement locomotive script.

NOTICE: THIS ENTIRE REPLACEMENT LOCOMOTIVE SCRIPT WORKS AS AN INJECTOR. IT DOES NOT REPLACE THE ORIGINAL LOCOMOTIVE SCRIPT (EVEN IF IT APPEARS THAT WAY). WE'RE NOT REPLACING THE ORIGINAL LOCOMOTIVE SCRIPT, WE'RE SIMPLY ADDING OUR OWN FUNCTIONS TO THE ALREADY EXISTING “BASE” CODING OF THE LOCOMOTIVE, HENCE WE HAVE TO “REQUIRE” THE “BASE” CODING IN ORDER TO BUILD UPON/EXTEND IT WITH OUR OWN FUNCTIONS/FEATURES. IT IS IMPORTANT TO UNDERSTAND THAT FOR THIS TO WORK, YOU EITHER HAVE TO HAVE ACCESS TO THE ORIGINAL LOCOMOTIVE SCRIPT (UNCOMPILED) OR BE AWARE OF THE FUNCTION NAMES ASSOCIATED TO SAID SCRIPT.

SINCE FOR THE MOST PART YOU WON'T HAVE ACCESS TO THE ORIGINAL, UNCOMPILED LOCOMOTIVE SCRIPTS NOR BE AWARE OF THE FUNCTION CALLS USED IN THOSE SCRIPTS, WE UNFORTUNATELY CAN ONLY “FILTER” THE ORIGINAL LOCOMOTIVE SCRIPTS FOR THE MOST COMMONLY USED FUNCTION CALL NAMES AND HOPE THAT THEY IN-FACT MATCH. BELOW IS A LIST OF ALL COMMONLY USED FUNCTION CALL NAMES. AS LONG AS THE LOCOMOTIVE YOU'RE INTENDING TO EQUIP WITH OUR *EQUIPMENT DEFECT DETECTOR* CAPABILITY IS USING ANY (AND NOT OTHER) OF THE FUNCTION CALLS BELOW, THIS TECHNIQUE WILL WORK.

MOST COMMON FUNCTION CALL NAMES *(FOR LOCOMOTIVE SCRIPTS ONLY)*

- *Initialise()*
- *OnControlValueChange()*
- *Update()*
- *OnConsistMessage()*
- *OnCustomSignalMessage()*
- *OnSave()*
- *OnResume()*

Provided you're skilled enough with LUA, you can obviously also incorporate our modifications into your existing locomotive script without having to “require” the original locomotive script.

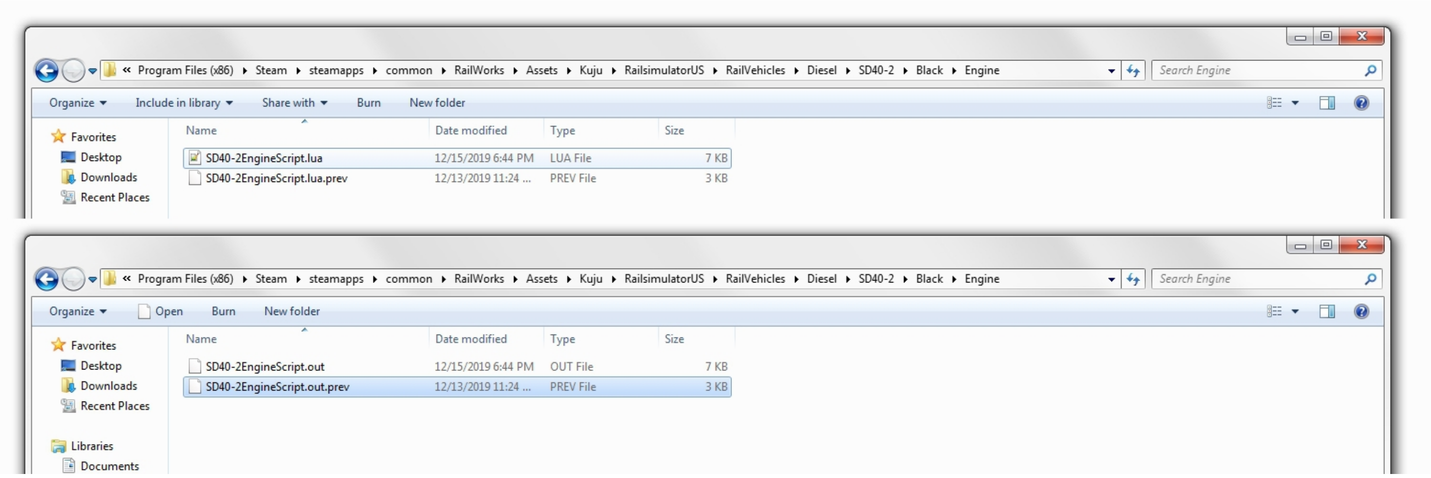
> *NOTE: DO NOT ATTEMPT TO EQUIP ANY ROLLING STOCK OR LOCOMOTIVES BROUGHT TO YOU BY JOINTEDRAIL/SEARCHLIGHT SIMULATIONS WITH EQUIPMENT DEFECT DETECTOR CAPABILITIES AS THAT WILL NOT WORK. WE WILL PROVIDE UPDATES WHICH WILL ENABLE THE FEATURE ON ALL OUR LOCOMOTIVES/ROLLING STOCK IN TIME.*

LOCOMOTIVE SCRIPT MODIFICATION

The final step (before we move onto the rolling stock) is now to check the file extension of our replacement locomotive script. Going back to page 3, we found out that our original engine script “SD40-2EngineScript.lua” was an uncompiled, LUA source file. If this is also the case with whatever locomotive script you're performing the changes on, you're done at this point, HOWEVER, should the original locomotive script file be a compiled LUA source file “SD40-2EngineScript.out” (as an example), you have to rename the extension from our replacement locomotive script from .lua to .out (so it matches the original) once you're done editing it.

There's no need to actually compile our replacement locomotive script at this point, renaming the extension will work just fine and have no impact on the game's capability to read the file's contents.

Your folder should now look similar to the image below. The above example for when you started out with an uncompiled LUA source file and the bottom example for when you started with a compiled LUA source file.


« Program Files (x86) ▸ Steam ▸ steamapps ▸ common ▸ RailWorks ▸ Assets ▸ Kuju ▸ RailsimulatorUS ▸ RailVehicles ▸ Diesel ▸ SD40-2 ▸ Black ▸ Engine
Search Engine
Organize ▾
Include in library ▾
Share with ▾
Burn
New folder
Favorites
Desktop
Downloads
Recent Places

Name	Date modified	Type	Size
SD40-2EngineScript.lua	12/15/2019 6:44 PM	LUA File	7 KB
SD40-2EngineScript.lua.prev	12/13/2019 11:24 ...	PREV File	3 KB

« Program Files (x86) ▸ Steam ▸ steamapps ▸ common ▸ RailWorks ▸ Assets ▸ Kuju ▸ RailsimulatorUS ▸ RailVehicles ▸ Diesel ▸ SD40-2 ▸ Black ▸ Engine
Search Engine
Organize ▾
Open
Burn
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents

Name	Date modified	Type	Size
SD40-2EngineScript.out	12/15/2019 6:44 PM	OUT File	7 KB
SD40-2EngineScript.out.prev	12/13/2019 11:24 ...	PREV File	3 KB

With all of the above complete, we can now move onto the rolling stock script modification (next page).

ROLLING STOCK SCRIPT MODIFICATION

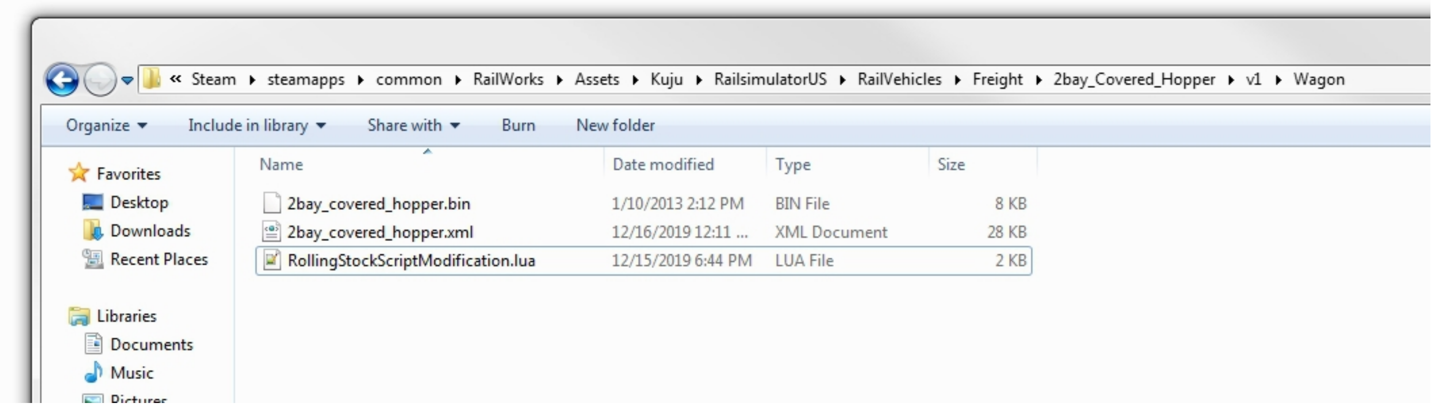
ROLLING STOCK SCRIPT MODIFICATION

NOTE: THE FOLLOWING STEPS NEED TO BE PERFORMED ON EVERY PIECE OF ROLLING STOCK/DLC INDIVIDUALLY (PROVIDED THE ROLLING STOCK IN QUESTION IS NOT ALREADY RUNNING ON A COMMON ROLLING STOCK SCRIPT OR ELSE).

To get going, you'll need to find the rolling stock's binary (.bin) file. For the example below we'll be using the Kuju 2bay_Covered_Hopper found under (Railworks/Assets/Kuju/RailsimulatorUS/RailVehicles/Freight/2bay_Covered_Hopper/v1/wagon).

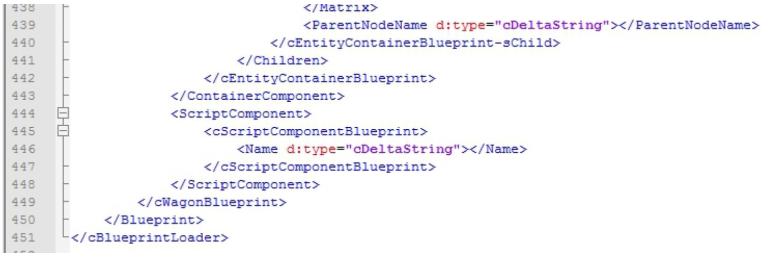
Repeat the steps from page 1 to 3 to retrieve the rolling stock's .bin file (if you haven't already) and again run it through the “serz.exe” located in your main railworks directory. That will create an .xml version of the original .bin file.

Before we continue, copy or drag the “RollingStockScriptModification.lua” file into the same directory as your rolling stock's newly created .xml file.


« Steam ▸ steamapps ▸ common ▸ RailWorks ▸ Assets ▸ Kuju ▸ RailsimulatorUS ▸ RailVehicles ▸ Freight ▸ 2bay_Covered_Hopper ▸ v1 ▸ Wagon
Organize ▾
Include in library ▾
Share with ▾
Burn
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures

Name	Date modified	Type	Size
2bay_covered_hopper.bin	1/10/2013 2:12 PM	BIN File	8 KB
2bay_covered_hopper.xml	12/16/2019 12:11 ...	XML Document	28 KB
RollingStockScriptModification.lua	12/15/2019 6:44 PM	LUA File	2 KB

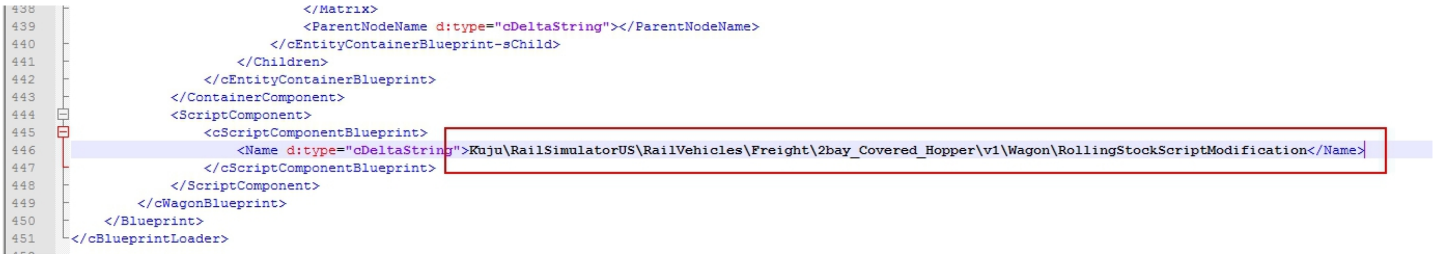
You then have to open that .xml file (we're using Notepad++ in our example) and scroll all the way to the bottom. The “ScriptComponentBlueprint” section should be blank (by default).



```
                            </Matrix>
                            <ParentNodeName d:type="cDeltaString"></ParentNodeName>
                        </cEntityContainerBlueprint-sChild>
                    </Children>
                </cEntityContainerBlueprint>
            </ContainerComponent>
            <ScriptComponent>
                <cScriptComponentBlueprint>
                    <Name d:type="cDeltaString"></Name>
                </cScriptComponentBlueprint>
            </ScriptComponent>
        </cWagonBlueprint>
    </Blueprint>
</cBlueprintLoader>
```

ROLLING STOCK SCRIPT MODIFICATION

You'll have to manually enter the full path to the “RollingStockScriptModification.lua” into the script component section of the rolling stock's blueprint file.



```
                            </Matrix>
                            <ParentNodeName d:type="cDeltaString"></ParentNodeName>
                        </cEntityContainerBlueprint-sChild>
                    </Children>
                </cEntityContainerBlueprint>
            </ContainerComponent>
            <ScriptComponent>
                <cScriptComponentBlueprint>
                    <Name d:type="cDeltaString">Kuju\RailSimulatorUS\RailVehicles\Freight\2bay_Covered_Hopper\v1\Wagon\RollingStockScriptModification</Name>
                </cScriptComponentBlueprint>
            </ScriptComponent>
        </cWagonBlueprint>
    </Blueprint>
</cBlueprintLoader>
```

NOTE: THE SCRIPT EXTENSION NAME SHOULD NEVER BE INCLUDED IN THE SCRIPT COMPONENT BLUEPRINT PATH.

Save and close the file. Next you have to delete the original .bin file and once done, drag the .xml file back into the “serz.exe”. It will compile the .xml file back into a .bin file. Our modified .bin file now has our rolling stock script referenced and is ready to be used in-game.

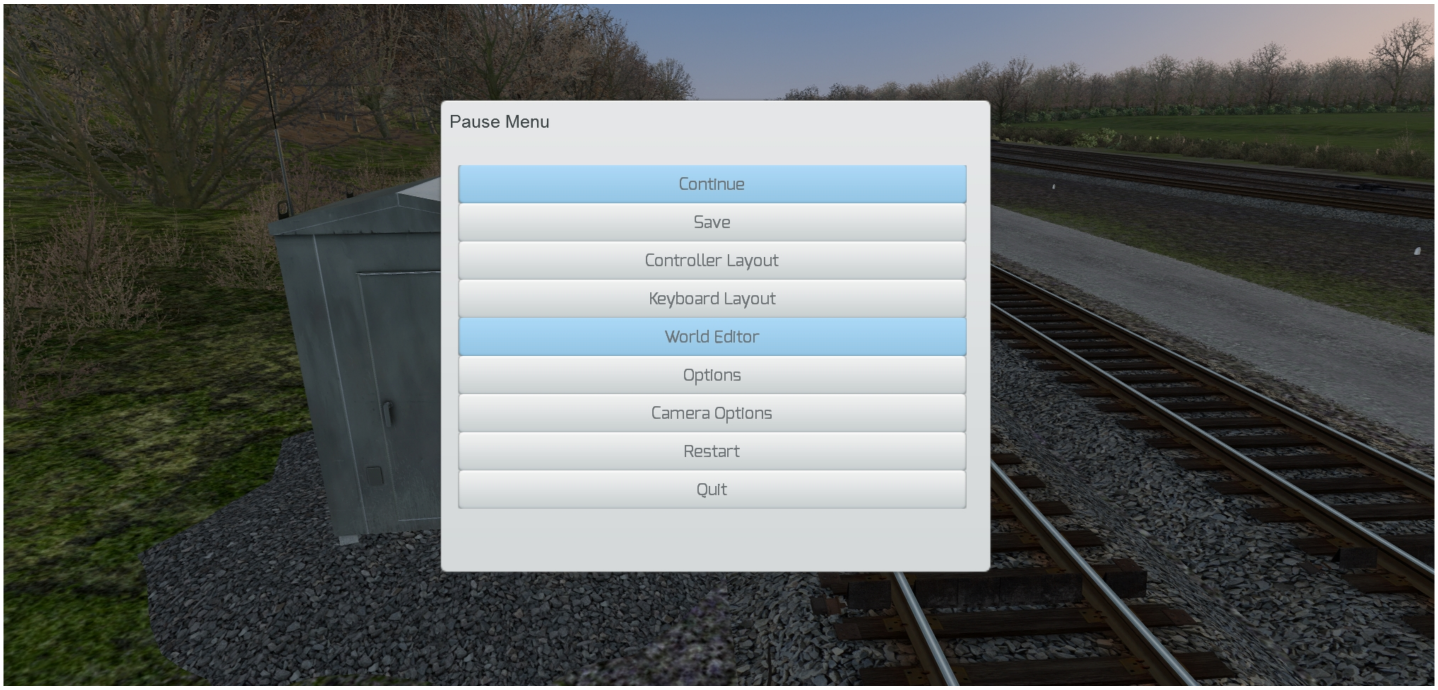
On the next page we'll take a look at the equipment defect detector itself and how to set it up in-game.

SET-UP INSTRUCTIONS IN-GAME

SET-UP INSTRUCTIONS IN-GAME

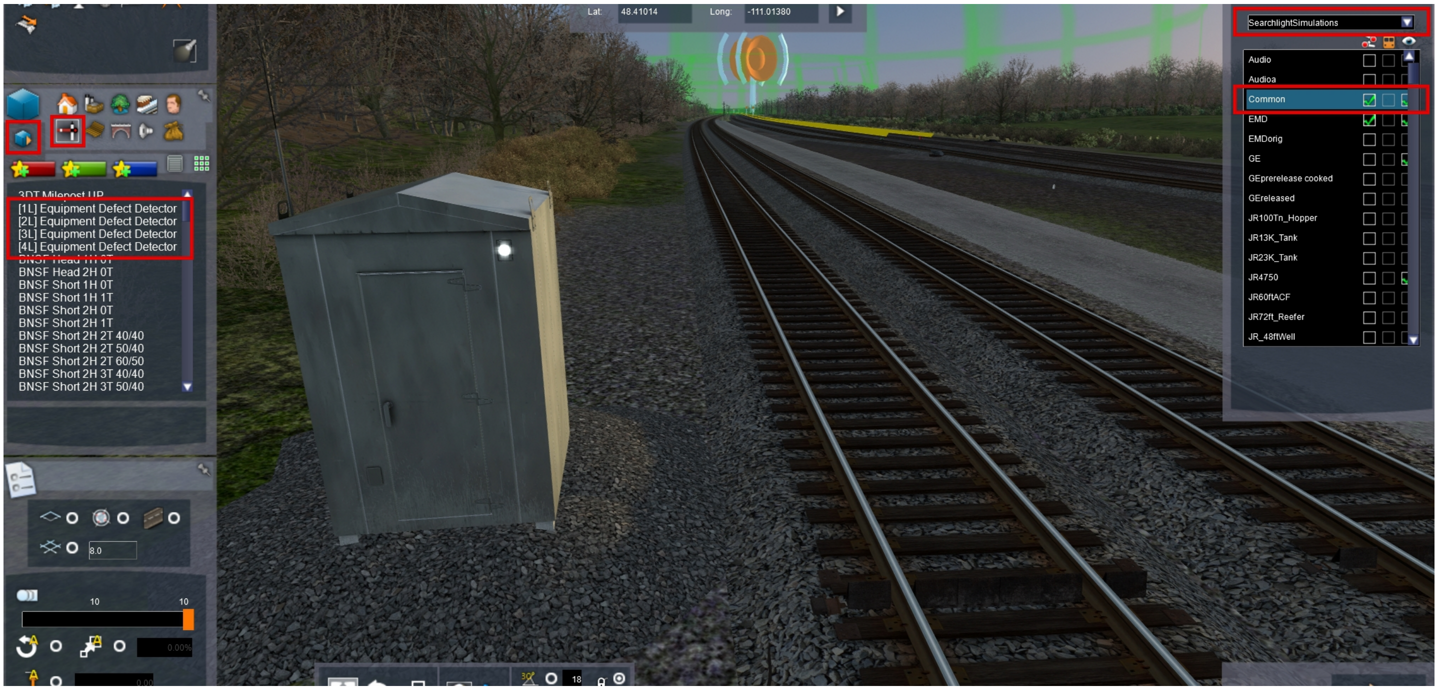
HOW TO PLACE AN EQUIPMENT DEFECT DETECTOR IN-GAME

In-game hit “escape” to bring up the pause menu and enter the world editor.



SET-UP INSTRUCTIONS IN-GAME

Next you'll have to open the assets filter. You do so by clicking on the small blue cube with the orange arrow. This will bring up the assets flyout on the right side, listing every DLC provider you have content off installed. Make sure to select “SearchlightSimulations” as your provider (first red box on the right side in the picture below) and look for “Common” as the product folder. It should appear at the very top. Tick both boxes (as shown in the picture below) and head back to the left side of your in-game screen.



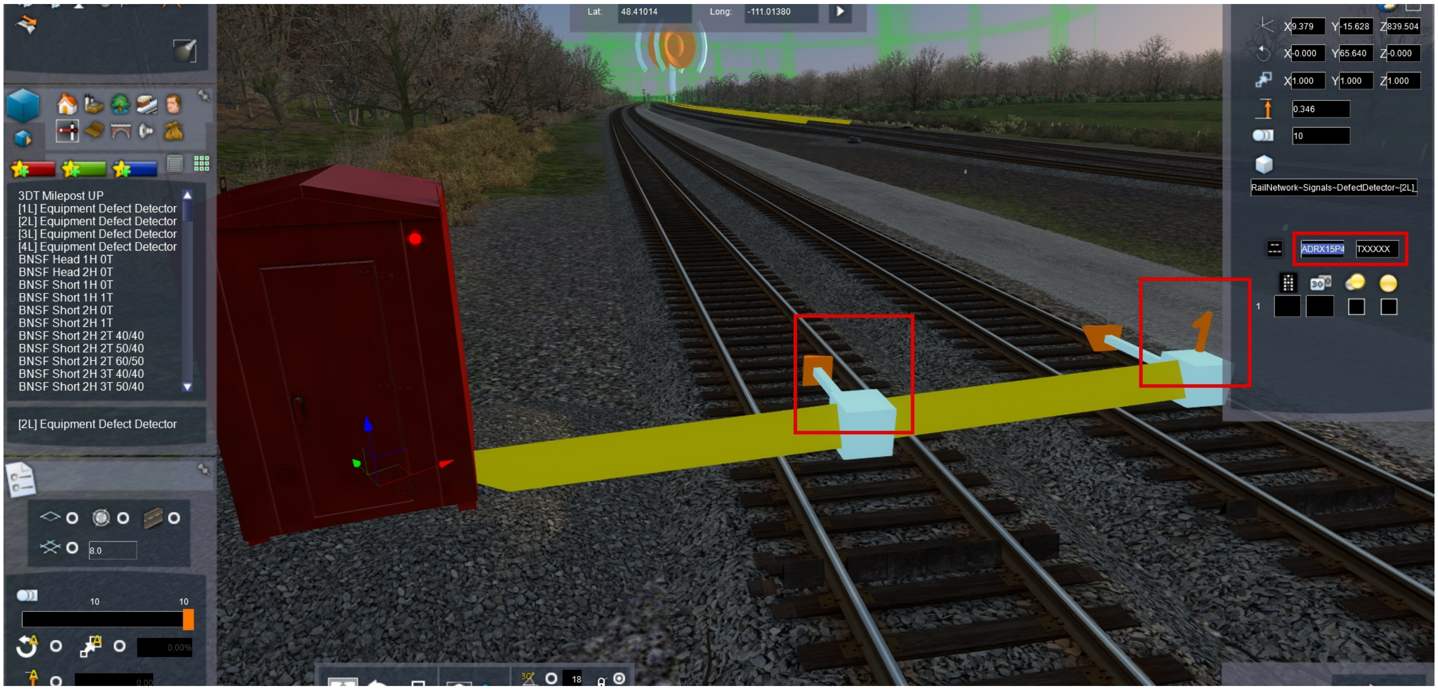
Our equipment defect detectors can be found under the “Track Infrastructure” section in the world editor (the small signal symbol) and will be listed as [1L]Equipment Defect Detector, [2L]Equipment Defect Detector etc. The [1L] prefix “1 LINK” etc indicates the available link count of the detector. For a single track (one link), you have to use the [1L] type, for two tracks (two links) the [2L] etc. . To properly place an equipment defect detector down, select the type and place down the PTMW box at your desired location by pressing (left mouse button). With the PTMW box still highlighted, you'll notice a blue arrow with an orange tip and a yellow link attached back to the PTMW box pop up.

Hover over the track you want to assign the equipment defect detector on and hit the left mouse button once again. This will assign the equipment defect detector to that specific track you've picked and will unhighlight the PTMW box.

If you chose the [1L] type, you'll only have to assign one link to your track since the equipment defect detector will only report from that single track. If you were to have two tracks and you wish to have an equipment defect detector report for both these tracks, you then would have to choose the [2L] type. When doing so, you'd be prompted to assign not one but instead two track links to each track individually. Same applies for the [3L] and [4L] track types respectively.

Furthermore, you'll find the actual equipment defect detector hardware to place down on your track once again under the “Track Infrastructure” section. It is labeled as “Equipment Defect Detector” in the list.

SET-UP INSTRUCTIONS IN-GAME


Lat: 48.41014 Long: -111.01380

X 9.379 Y -15.628 Z 839.504
X -0.000 Y 65.640 Z -0.000
X 1.000 Y 1.000 Z 1.000
0.346
10
RailNetwork~Signals~DefectDetector~[2L]_
ADRX15P4 TXXXXX

Notice how there's a “1” floating above the second track link but not the first one. In-game, the first track link will always be assigned the index 0, the second track link will be assigned index 1 etc. however only a track index greater than 0 will be displayed above the actual track link in-game.

If you're on track 1, it's index is 0, track 2's is index 1 etc. It's an easy way to later identify or troubleshoot the equipment defect detector.

Once you've successfully placed an equipment defect detector down in-game, it's now time to properly configure it for reports. In the picture above you'll notice two separate input boxes on the right side flyout. Each one of them serves a specific function.

The first input box (11 digit string) takes care of the railroad this particular equipment defect detector is assigned to, the milepost position and milepost specifics (if there are any). Below is a list of (currently) all available railroad IDs you can assign to your equipment defect detector. The railroad ID is a (4 digit) long string of specific characters for each railroad.

- BNSF	*Burlington Northern Santa Fe*
- CPRX	*Canadian Pacific*
- NSXX	*Norfolk Southern*
- KCSX	*Kansas City Southern*
- CSXT	*CSX Transportation*
- ICXX	*Illinois Central*
- BNXX	*Burlington Northern*
- UPXX	*Union Pacific*
- SPXX	*Southern Pacific*
- ADRX	*Adriana County*

SET-UP INSTRUCTIONS IN-GAME

You can assign up to 4 digit (plus decimal) long milepost IDs to our equipment defect detectors. It is very important that you understand the correct way of entering your desired milepost ID or else the equipment defect detector will not work.

The milepost ID is a (7 digit) long string of numbers or specific characters. Only [1 through 9, N, P, X] are valid entries for the milepost ID. If you type anything else, it will break the equipment defect detector thus rendering it useless.

Here's a few examples on what to enter for the milepost ID in order to achieve a specific detector readback.

Valid entries:
Example 1. ".. Detector, Milepost 15 Point 6 .." = 15P6XXN
Example 2. ".. Detector, Milepost 206 Point 2 .." = 206P2XN
Example 3. ".. Detector, Milepost 1 Point 5, X .." = 1P5XXXX
Example 4. ".. Detector, Milepost 1206 Point 3, X .." = 1206P3X

Invalid entries:
Example 1. ".. Detector, Milepost 101 Point 6 .." = 1X1P60X
Example 1. ".. Detector, Milepost Point 15 Point 3 .." = P15P3XN

For the first 6 digits, an "X" indicates a blank field. Even if your milepost is not 6 digits long, you still have to fill out the gaps in the 7 digits long string of numbers and letters or else the game would attempt to read a "nil" or invalid argument/value thus breaking it. The letter "P" stands for POINT and can only be assigned for the first 6 digits of the available 7 digit string.

The 7th or last digit of the 7 digit long string is your milepost specifics suffix. IRL mileposts at times will be assigned a specific suffix after the actual milepost ID. In-game you have the same option by either typing an "X" or "N" for the last digit. If you type in "X" the detector will report ".. Milepost […] X, Track/Main...". If you don't need any milepost specifics, simply type in "N" for the last digit of the 7 digit long string.

In conclusion, you should now have the first input box properly filled up with a total of 11 characters:

RR CODE (4 digits) + MILEPOST ID (7 digits)

The second input box (6 digit string) will decide whether the equipment defect detector is reporting a "Main" track or just a plain "Track", whether or not it will report the speed, ambient air temperature, car count or train length. Since every detector IRL varies between what and what not it reports, we gave the user the ability to have full control about what it reports in-game as well.

The following characters are available for input (digit by digit in order left to right):

1st digit: "M" = Main, "T" = Track, "X" = Unavailable (skip report)
NOTE: The [1L] equipment defect detector type will always skip the track type report, no matter the user input.

2nd digit: "X" = Currently disabled (reserved for CNs "North-, East-, South- and Westbound" report)
3rd digit: "S" = Speed report true, "X" = Speed report false
4th digit: "T" = Temperature report true, "X" = Temperature report false
5th digit: "C" = Total car count report true, "X" = Total car count report false
6th digit: "L" = Total length report true, "X" = Total length report false

SET-UP INSTRUCTIONS IN-GAME

Here's a few examples on what to enter for the second input box in order to achieve a specific detector readback.

Valid entries:
Example 1.
MXSTXX = Report Track Type: “MAIN” Track, Report Train Speed (YES), Report Ambient Air Temperature (YES), Report Train Total Car Count (NO), Report Train Length (NO)

Example 2.
MXSXXL = Report Track Type: “MAIN” Track, Report Train Speed (YES), Report Ambient Air Temperature (NO), Report Train Total Car Count (NO), Report Train Length (YES)

Example 3.
TXSXXX = Report Track Type: “TRACK”, Report Train Speed (YES), Report Ambient Air Temperature (NO), Report Train Total Car Count (NO), Report Train Length (NO)

Example 4.
MXSXXL = Report Track Type: “MAIN” Track, Report Train Speed (YES), Report Ambient Air Temperature (NO), Report Train Total Car Count (NO), Report Train Length (YES)

Example 5.
MXSTCL = Report Track Type: “MAIN” Track, Report Train Speed (YES), Report Ambient Air Temperature (YES), Report Train Total Car Count (YES), Report Train Length (YES)

Invalid entries:
Example 1.
MNSTXX = Invalid because 2nd argument does not equal “X”

Example 2.
MXXXLC = Invalid because 5th and 6th argument is swapped

Now a full example of what report the first and second box combined will result in-game:

First input box: BNSF146P1XN   Second input box: MXSXXX

“BNSF DETECTOR MILEPOST ONE FOUR SIX POINT ONE MAIN [..] NO DEFECTS REPEAT NO DEFEACTS TOTAL AXLE […] SPEED […] OUT”

NOTE: Train speed is measured at the first and last axle of your train and the equipment defect detector then determines the average train speed on report. If your average train speed drops below 17MPH, the equipment defect detector will report “TRAIN TOO SLOW”.

IMPORTANT INFORMATION

INCONSISTENT AUDIO READBACK IN-GAME VS REAL LIFE

The base audio recordings that power the equipment defect detector come straight from factory. Unfortunately we're missing certain audio recordings for each specific railroad. All railroad IDs for instance except Norfolk Southern have been “pieced” together with audio files form a-z. This is the reason they sound different in-game than they do IRL even though it's the same voice.

Same applies to the “Equipment Defect Detector” and “Niner” instead of “Nine” report on CSXT detectors.

Amongst other things, we're also missing the proper “Mile”, “North-, East-, South- and West-bound” audio files along with the “Integrity Failure” report, meaning the only limitations we face are the lack of specific audio recordings.

Otherwise we would have included Canadian National in our detector list too but since we're missing specific audio recordings necessary to accurately replicate CNs equipment defect detectors, we unfortunately had no choice but to skip it for the time being.

Of course we'll upgrade our equipment defect detector should more audio recordings from the factory become available to us. If you have access to the actual, factory recordings which match the same detector voice, feel free to get in touch with us.

TROUBLESHOOTING

DETECTOR IS NOT REPORTING ANYTHING

Depending on your axle count, the detector will take anywhere from 10-30 seconds to process the data and report it back.

If you notice you're detector has not reported anything (even after 30 seconds), check the following folder located in: *“RailWorks\Assets\SearchlightSimulations\Common\RailNetwork\Signals\DefectDetector”* and see whether you can spot a file named “EDD_LAST_TRAIN_INFO.lua”.

If no such file is present, it either indicates an issue with your modified engine script or you've attempted to cross the equipment defect detector with just a single locomotive and no other locomotive(s) trailing or freight car(s) attached to your train. The equipment defect detector will not pick up a light engine move.

NOTE: MULTIPLE EQUIPMENT DEFECT DETECTORS CAN NOT SHARE THE SAME BLOCK (TRACK PIECE). PLACING MULTIPLE EQUIPMENT DEFECT DETECTORS ON THE SAME BLOCK WILL RENDER ALL OF THEM UNFUNCTIONAL.

CREDITS AND AFTERWORDS

We'd like to thank our friends over JointedRail for providing us with the factory audio recordings to power our equipment defect detectors. Their audio recordings add a lot to the overall immersion/experience in-game and certainly would have not been possible otherwise.

Make sure to give them a like on their facebook page or show support otherwise by checking our their excellent product selection for Trainz!

www.jointedrail.com
www.facebook.com/jointedrail

For proper support inquiries related to this product and or others, please use our support form on the our website or directly contact us at support@searchlight-simulations.com . Also make sure to subscribe to our newsletter if you haven't already!

Stay updated!

www.store.searchlight-simulations.com

www.facebook.com/SearchlightSimulations