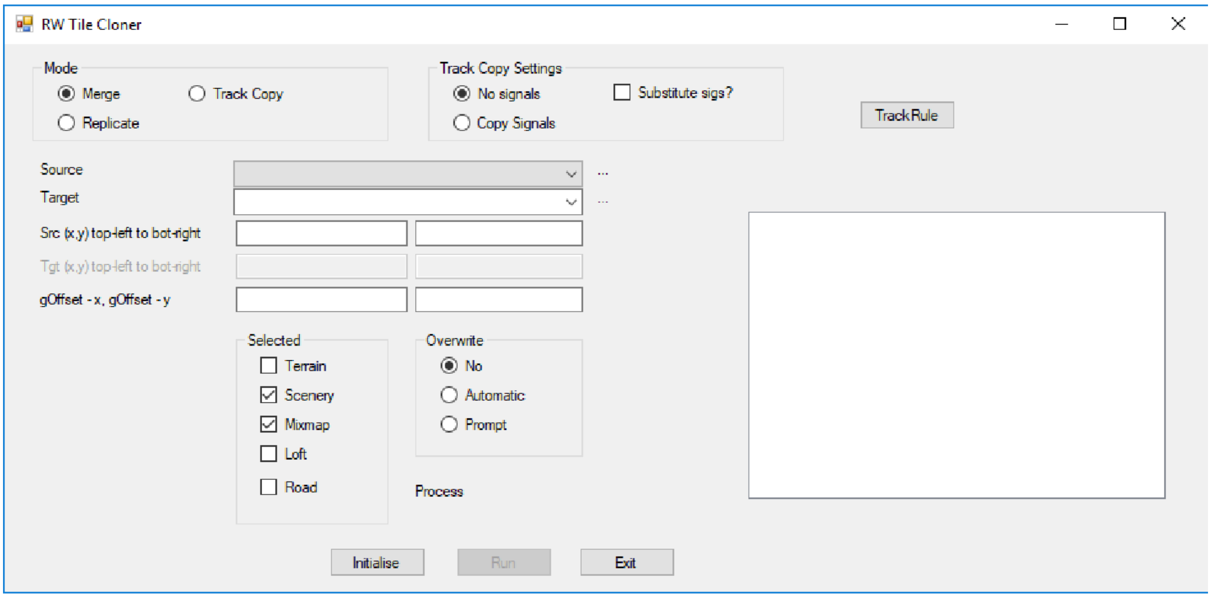
RWKlon

A not-manual for use

RWKlon is a rough utility designed to do several jobs that assist in cloning parts of a route or merging two different routes together. It is not releasable – the UI is too rough and overly-specific and the coding is not up to my own standards in several places. Used at your own risk.

You'll need VB express (unless you fancy upgrading to a later version) and some basic VB knowledge.

As usual – take backups of everything before you start, during the process after etc etc.



RW Tile Cloner

Mode
● Merge ○ Track Copy
○ Replicate

Track Copy Settings
● No signals ☐ Substitute sigs?
○ Copy Signals

Source
Target
Src (x,y) top-left to bot-right
Tgt (x,y) top-left to bot-right
gOffset -x, gOffset -y

Selected
☐ Terrain
☑ Scenery
☑ Mixmap
☐ Loft
☐ Road

Overwrite
● No
○ Automatic
○ Prompt

Process

What does it do?

Merge - This copies tiles in the categories in the 'selected' frame from the source to the target route. It adjusts the tile offsets to accommodate the offset (gOffsetx, gOffsetY) from one route to the other. It doesn't change anything else in the tiles (except for Mixmap tiles which will have their texture set mapped according to an internal table).

Track Copy – this does the same job for the track tiles and tracks.bin – BUT – see below for all the fun that happens after that. This includes options to do with or without signals – and also a crude WIP converter to convert any signals that it recognises from CLS into semaphore (not finished).

Replicate – this takes a block of tiles (in the categories in 'selected') and clones them repeatedly over the section of the same route. It is used for replicating large swathes of fields etc that get very boring to do manually.

Before you start

Copy Serz.exe into C:/temp – and make sure that you have full access to c:\temp. This is used as a working folder.

Do an initial; test -find a 2x2 area of occupied tiles in your source route (each one with at least one scenery item on it) - and then test that into a dummy route or an empty space in your desired target route. That should work without error unless there is some other issue.

Think about what you are going to do. I always recommend doing scenery etc before doing the track as it is easier to see if you are copying what you wanted and where you are putting it. The program uses hard-coded arrays so it has upper limits (10,000 tiles and 50,000 track items) – I have increased these (16-04-19 - they were an order of 10 smaller) but haven't tested – should work – but I also suggest you run the tool in a few chunks rather than all at once unless you are cloning a very small route.

Merge

Select 'Merge' as the option

Select the route GUID for the source of the tiles and the target (the route that will receive them) – NB – I have hard-coded 3 specific routes for my own convenience (in frmMain_Load) – you will need to alter according to your own needs (this also contains the hard-coded path to railworks – may also need amending).

Set the tiles refs for 'src (x,y)…' – these are the refs of the top left and bottom right tiles that will be cloned (+000001,+000004 etc)

Set gOffsetx, gOffsetY – these give the offset in tiles between the coordinates in the source and target routes – ie the numbers here will be added/subtracted from the source tile refs so that they fit in the target route. You need to work this out – take a convenient location with a match on both routes – get its tile refs and then work out an offset. (**for the offsets you subtract the source from the desired target - ie +105 minus -68 = +173.**)

Set the 'Overwrite' choice – none (tiles that already exist in the target route aren't overwritten), Prompt (it prompts you to say whether each tile should be overwritten) or Automatic – here it just overwrites anything that was there.

Hit ' Initialise' – this reads and validates your selections – if ok it throws up a list of the new tiles to be created.

Hit 'Run' – this then does the deed – good luck! This will list what it is doing in the same box – you may get the 'source not present' message – this is fine for any empty tiles (and you will get lots of those in a large rectangle with the route tiles meandering through it) - it is only a problem if you get that message for ALL tiles.

Post-Merge activities

You WILL have to fiddle and fudge at the join – the aim is to blend the two sections together so that no one can see the join.

Fun with Mixmap

If your two routes use the same texture maps then yippee. Otherwise you have to either add all of one set of textures to the other one or select a 'nearest fit' and then map the source texture to an established texture in the target. You do this by comparing the colours in game – then identifying the ordinal refnbr for the texture and editing the table in this routine:

```
Function Mixmap_Convert
```

If you don’t know how mixmaps are defined in the game then have a look at the code and a little dig around – it’s not hard – basically there is an xml table listing each of the textures used (with 4 textures – one for each season) – and the ordinal number is used in the mixmap tiles as a reference.

See the attached ‘RW-Cloner-mixmap-mapping.xlsx’ for an example that I used to work out what to do.

Addendum:
Re mixmaps. You work through the terrain.bin in the target route and write down each texture on a spreadsheet with ordinal numbers starting from 1. You then write down the texture names for your source route in the next column. And then you work out which 'original' texture is closest to the texture in your 'source' column - and write the original/target number into column three. so you end up with:
01 green grass dry grass 02
02 dry grass stone 12
..
12 stone green grass 01

you then copy the target number in sequence into the table. This tells the routine to change the texture number in each cell of the source route tile into the number that corresponds with the equivalent texture in the target route.
You may find some textrues in your source route that don't have an equivalent in the target route - so you can copy and paste those definitions from the source bin to the target bin (at the end) and they use the next ordinal number in the conversion process.
E.g. you need 'aardvark texture' but it isn't in the target route. So you add aardvark texture to the end of the target terrain.bin (at position 46) and add it to your mapping table with the number 46.

Texture - in a proper tool this would be an external file - but since it's not it is hard-coded! The code sits in 'Function Mixmap_Convert' and uses the target part of a mapping table. To make sense of it see this file: https://drive.google.com/file/d/1QJ3B9I ... sp=sharing

Basically each route has a set of textures with an identifying number (01, 02 etc) which is used to state what texture each chunk of the tile is painted. Sadly they can all use the same number to represent a different texture...The textures are held in a binary (Assets\cunningn\Cambs\Environment\Terrain\CambsTerrain.bin is the one for my route) pointed to by the template?? (can't remember - you may need to dig around). The textures are defined by ordinal position within the terrain.bin - so you need to map out the two sets of textures and then decide which 'greengrass' textureis your 'master and map accordingly. It would have been FAR better had they used an absolute number and made them all unique....

Replicate

Setup in a similar way to ‘merge’ – I use it with tiles from the same route but could be used from a different route.

NB – It doesn’t adjust heights for scenery or lofts – so if your terrain is bumpy then either just do the mixmap or do wide-area selects followed by ‘j’ followed by some fettling!

Track Copy

Originally I just did the other tiles and added the new track manually. The ‘track copy’ option takes the manual effort away – but there is still work to be done.

Setup is the same as for ‘merge’ and it runs the same way. I always recommend doing it separately AFTER doing the scenery (and taking a backup…)

It should all work automatically now – but check the route with rwtools afterwards and see the appendix if you need to fix anything….

I haven’t done the level crossings or fuel points – could be added – wasn’t relevant to me…

Big routes have issues

It seems that as routes (tracks.bin I guess) get larger they start throwing up odd extra errors in Logmate (I thought this was due to using the cloner – but AndiS gets it on other routes too):

-always gives a set of ‘ribbon id not found’ errors - but these vary according to location - often hundreds - these and the assert errors slow down startup but have no other effect??
-when the ribbon-id is all zeroes then this suggests that it is looking for one and can’t find it
-when the ribbon id has a value this is present only in the tracks.bin as an actual ribbon and then as a connector -I have tried deleting the connector as a test for one case without either success or harm…. AndiS has also encountered and looked into this – neither of us has a solution – but the route seems to run ok anyway..

Track rule missing

```
[RunTimeError 15:34:05] Track Rule missing: ////
[RunTimeError 15:34:05]
[RunTimeError 15:34:05] ASSERT(mTrackRule != 0);
[RunTimeError 15:34:05]
[RunTimeError 15:34:05] Network::iTrackNetworkTrackRule::cPropertyValue::setTrackRule()
[RunTimeError 15:34:05]
[RunTimeError 15:34:05] C:\build\CoreRelease\Code\\PublicInterfaces\iTrackNetworkProperties.d.h
: 84
```

resolved this by identifying all places where trackrule was missing then editing (and then just did a mass edit which was quicker) - this reduced load time by 5-6 seconds?

Mixmap error

Reported in Logmate:

```
Verify failed:
cGBuffer.cpp : 517
Expression: (count = mSurfaces[eSurfaceAccessibleDepth]->Release()) == 0
```

-probably due to bad mixmap settings – on re-applying a good set of mixmaps this went away.?

What Else?

There are numerous ways in which this utility could be made ‘better’

Add a specific offset difference on a per tile basis to get a more accurate match between source and target routes. This could be done in all 3 planes. However – this would be a fair bit of work and in reality none of my 3 merges have really needed it – the maximum discrepancy (in theory) could be 500 metres – but it’s more likely to be a couple of hundred – and you can ‘lose’ this in the manual join. None of my joins were perfect (though the Peterborough one wasn’t at all bad) – but none is noticeable.

North-South divergence affects multiple East-West joins. The game doesn’t handle the curvature of the earth – so two routes that head North/South from different origins will veer away from each other as they go further North. If you need to make multiple E-W joins then you WILL have to add some extra track to bridge the gap. It’s best to choose a mid-point in the N-S axis and accept tweaks for E-W joins either above or below this.

Fix the damn blueprint xml bug!

It could be made for ‘consumer-friendly’ by taking dynamic lists of available routes (and including their names) – but why bother?

The mixmap stuff could be slicker – but you do it once so wtf.

The track stuff is the ‘biggie’ – multiple issues here;

- Fix the blueprint bug here too
- Enhance the signalling conversion – I have made a crude attempt (converting CLS to semaphore) so you can follow and do your own if desired
- Fix the broken links bug – more cleverness needed here – should be do-able.

Appendix – using rwtools to fix broken bits

Scenery or loft tiles can get a bad piece of xml created – this is an artefact of the Vb XML writer AFAIK. It needs fixing but I haven’t got round to it. Instead I use RWTools to scan for the broken XML and replace with good stuff – this will affect multiple tiles so it is worth doing.

-SCENERY:

```
                              <iBlueprintLibrary-cAbsoluteBlueprintID>
                                     <BlueprintSetID>
                                            <iBlueprintLibrary-cBlueprintSetID>
                                                   <Provider d:type="cDeltaString">
      </Provider>
                                                   <Product d:type="cDeltaString">
      </Product>
                                            </iBlueprintLibrary-cBlueprintSetID>
                                     </BlueprintSetID>
                                     <BlueprintID d:type="cDeltaString">
    </BlueprintID>
                              </iBlueprintLibrary-cAbsoluteBlueprintID>
```

LOFT/ROAD:

```
<iBlueprintLibrary-cAbsoluteBlueprintID>

<BlueprintSetID>

<iBlueprintLibrary-cBlueprintSetID>

          <Provider d:type="cDeltaString">
               </Provider>

          <Product d:type="cDeltaString">
               </Product>

</iBlueprintLibrary-cBlueprintSetID>

</BlueprintSetID>
                                                                                                              <BlueprintID

d:type="cDeltaString">
            </BlueprintID>

</iBlueprintLibrary-cAbsoluteBlueprintID>
```

REPLACE WITH:

```
<iBlueprintLibrary-cAbsoluteBlueprintID>
          <BlueprintSetID>
                    <iBlueprintLibrary-cBlueprintSetID>
                              <Provider d:type="cDeltaString"></Provider>
                              <Product d:type="cDeltaString"></Product>
                    </iBlueprintLibrary-cBlueprintSetID>
          </BlueprintSetID>
          <BlueprintID d:type="cDeltaString"></BlueprintID>
</iBlueprintLibrary-cAbsoluteBlueprintID>
```