RailCOMPLETE v1.0 Reference Manual

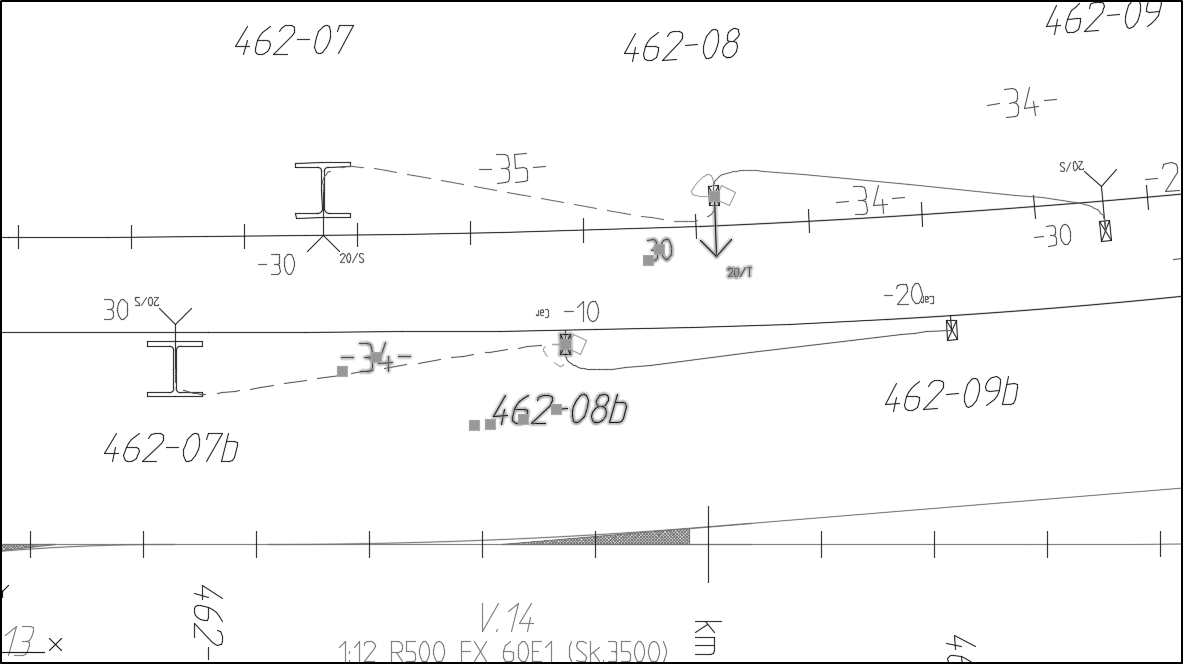


Table of Contents

[Welcome 9](#_Toc503345299)

[RailCOMPLETE is a CAD system plug-in 9](#_Toc503345300)

[RailCOMPLETE is a professional and versatile tool 9](#_Toc503345301)

[RailCOMPLETE is based on a safe data format 9](#_Toc503345302)

[RailCOMPLETE protects your engineering investment 9](#_Toc503345303)

[RailCOMPLETE genetics – Object Type Definitions 10](#_Toc503345304)

[RailCOMPLETE Agents 10](#_Toc503345305)

[RailCOMPLETE Patches 10](#_Toc503345306)

[The RailCOMPLETE Business Case 10](#_Toc503345307)

[Getting started 12](#_Toc503345308)

[Starting RailCOMPLETE 12](#_Toc503345309)

[RailCOMPLETE commands 12](#_Toc503345310)

[Start a new RailCOMPLETE document 13](#_Toc503345311)

[A quick example 14](#_Toc503345312)

[Main features 18](#_Toc503345313)

[Manage Objects 18](#_Toc503345314)

[Locate in Manage Objects 18](#_Toc503345315)

[Locate in drawing 19](#_Toc503345316)

[Add and remove columns 20](#_Toc503345317)

[Searching and filtering 20](#_Toc503345318)

[Using formula 23](#_Toc503345319)

[Set formula 24](#_Toc503345320)

[Auto-fill: Copy formula 26](#_Toc503345321)

[Auto-fill: Fill series 26](#_Toc503345322)

[Find and replace 27](#_Toc503345323)

[Create table from Object Manager 28](#_Toc503345324)

[Export to Excel 28](#_Toc503345325)

[Manage Alignments 29](#_Toc503345326)

[Manage Stages 29](#_Toc503345327)

[Manage Properties 30](#_Toc503345328)

[Manage Tables 32](#_Toc503345329)

[Q & A 34](#_Toc503345330)

[Speed issues 34](#_Toc503345331)

[Manage Stages 34](#_Toc503345332)

[Object Type Definitions 34](#_Toc503345333)

[Position 35](#_Toc503345334)

[LUA 35](#_Toc503345335)

[RailCOMPLETE dialog window is not visible 35](#_Toc503345336)

[RailCOMPLETE commands 36](#_Toc503345337)

[RC-ABOUT 36](#_Toc503345338)

[RC-APPROXIMATEPOLYLINEBYARCS 36](#_Toc503345339)

[RC-ASSISTCREATESWITCHESANDCONNECTIONS 37](#_Toc503345340)

[RC-ASSISTSETMILEAGEPARENTS 37](#_Toc503345341)

[RC-ATTACHXREF 38](#_Toc503345342)

[RC-ATTSYNC 38](#_Toc503345343)

[RC-AUDITALIGNMENT 38](#_Toc503345344)

[RC-AUDITALL 38](#_Toc503345345)

[RC-AUDITALLRELATIONS 38](#_Toc503345346)

[RC-AUDITUNIQUEIDS 38](#_Toc503345347)

[RC-BINDXREF 38](#_Toc503345348)

[RC-BREAKALIGNMENTS 38](#_Toc503345349)

[RC-BROWSECOMMANDS 38](#_Toc503345350)

[RC-CALCULATEFRESNEL 39](#_Toc503345351)

[RC-COPYANNOTATIONSTODRAWING 39](#_Toc503345352)

[RC-COPYBRAKINGCURVETOCLIPBOARD 39](#_Toc503345353)

[RC-COPYPOSITIONTOCLIPBOARD 39](#_Toc503345354)

[RC-COPYSELECTIONALONGPATH 39](#_Toc503345355)

[RC-CREATEALIGNMENT 39](#_Toc503345356)

[RC-CREATEAREA 40](#_Toc503345357)

[RC-CREATECONNECTION 41](#_Toc503345358)

[RC-CREATELAYERMAPPING 41](#_Toc503345359)

[RC-CREATEOBJECTTYPEDEFINITIONSMAPPING 41](#_Toc503345360)

[RC-CREATEPOINTOBJECT 41](#_Toc503345361)

[ALIGNMENT AND POSITION 42](#_Toc503345362)

[COORDINATE SYSTEM 42](#_Toc503345363)

[REFERENCE ALIGNMENT 42](#_Toc503345364)

[DIRECTION 42](#_Toc503345365)

[OBJECT PROPERTIES 43](#_Toc503345366)

[POSITION IN DRAWING VS. “REAL” XY POSITION XY 43](#_Toc503345367)

[ATTACHING OBJECTS TO EACH OTHER 43](#_Toc503345368)

[TEXT ATTRIBUTES 44](#_Toc503345369)

[3D VISUALIZATION 44](#_Toc503345370)

[RC-CREATETABLE 44](#_Toc503345371)

[RC-DETACHXREF 44](#_Toc503345372)

[RC-DISPLAYBRAKINGCURVEEBICAB700 44](#_Toc503345373)

[RC-DISPLAYSIGHTAREA 45](#_Toc503345374)

[RC-DISPLAYSIGHTLINE 45](#_Toc503345375)

[RC-DISPLAYTRAIN 46](#_Toc503345376)

[RC-DISPLAYTRAINENVELOPE 46](#_Toc503345377)

[RC-EDITPROPERTY 47](#_Toc503345378)

[RC-EDITSETTINGS 47](#_Toc503345379)

[RC-EDITTABLE 47](#_Toc503345380)

[RC-EXPORT3D 47](#_Toc503345381)

[RC-EXPORTALIGNMENTSTOLANDXML 48](#_Toc503345382)

[RC-EXPORTREFERENCEALIGNMENTDEFINITIONS 48](#_Toc503345383)

[RC-EXPORTROUTES 48](#_Toc503345384)

[RC-EXPORTTABLETOEXCEL 48](#_Toc503345385)

[RC-IMPORTALIGNMENTSFROMDWG 48](#_Toc503345386)

[RC-IMPORTALIGNMENTSFROMLANDXML 48](#_Toc503345387)

[RC-IMPORTREFERENCEALIGNMENTDEFINITIONSFROMXML 48](#_Toc503345388)

[RC-JOINALIGNMENTS 48](#_Toc503345389)

[RC-LOADOBJECTTYPEDEFINITIONSFROMDRAWING 48](#_Toc503345390)

[RC-LOADOBJECTTYPEDEFINITIONSFROMXML 49](#_Toc503345391)

[RC-LOADRIBBON 49](#_Toc503345392)

[RC-LOADSTYLEDEFINITIONSFROMXML 49](#_Toc503345393)

[RC-LOADSYMBOLLIBRARYFROMDWG 49](#_Toc503345394)

[2D SYMBOLS USUALLY CONTAIN GRAPHICS ON SEVERAL CAD LAYERS 50](#_Toc503345395)

[2D SYMBOL PARTS 50](#_Toc503345396)

[CHANGING 2D SYMBOL WHEN VARIANT IS CHANGED 51](#_Toc503345397)

[TEXT ATTRIBUTES IN 2D SYMBOLS 51](#_Toc503345398)

[RC-MANAGEALIGNMENTS 51](#_Toc503345399)

[RC-MANAGEOBJECTS 51](#_Toc503345400)

[OBJECT CATEGORY BROWSING 53](#_Toc503345401)

[AREAS 53](#_Toc503345402)

[SELECTED IN DRAWING 53](#_Toc503345403)

[SELECTION AND MARKING 54](#_Toc503345404)

[MODIFYING CELL VALUES 54](#_Toc503345405)

[FORMULAS 54](#_Toc503345406)

[LUA EDITOR 54](#_Toc503345407)

[REMOVING FORMULAS 55](#_Toc503345408)

[DRAWING ATTENTION TO A VALUE – WARNINGS AND CEHCK MARKS 56](#_Toc503345409)

[EXPORT OBJECT DATA TO EXCEL 56](#_Toc503345410)

[CREATING TABLES FROM THE OBJECT MANAGER 57](#_Toc503345411)

[RC-MANAGEPROPERTIES 57](#_Toc503345412)

[RC-MANAGESTAGESUSINGLAYERNAMES 57](#_Toc503345413)

[RC-MATCHPOSITION 58](#_Toc503345414)

[RC-MOVESELECTIONALONGPATH 58](#_Toc503345415)

[RC-MOVETEXTATTRIBUTES 58](#_Toc503345416)

[RC-OPENXREF 58](#_Toc503345417)

[RC-P000001-UPDATELABELS 58](#_Toc503345418)

[RC-P000002-UPDATETABLES 58](#_Toc503345419)

[RC-P000003-MOUNTING3D 58](#_Toc503345420)

[RC-P000004-FIXMILEAGE 58](#_Toc503345421)

[RC-P000005-FIXCONNECTIONS 59](#_Toc503345422)

[RC-P000006-CHANGEDATATYPEOF68SIGNALS 59](#_Toc503345423)

[RC-P000007-RENAMETRACKTOALIGNMENT 59](#_Toc503345424)

[RC-P000010-MOVEOFFSETFORMULAS 59](#_Toc503345425)

[RC-PEEKATOBJECTDEFINITION 59](#_Toc503345426)

[RC-PEEKATOBJECTTYPEDEFINTIONS 59](#_Toc503345427)

[RC-PEEKATTABLEDEFINITION 61](#_Toc503345428)

[RC-QUANTIZEPOSITION 61](#_Toc503345429)

[RC-QUERYOBJECT 61](#_Toc503345430)

[RC-QUICKSELECT 62](#_Toc503345431)

[RC-REFRESHOBJECTS 62](#_Toc503345432)

[RC-REFRESHTABLE 62](#_Toc503345433)

[RC-RELATIONUPDATEDEPTH 62](#_Toc503345434)

[RC-RELOADALLXREFS 62](#_Toc503345435)

[RC-RELOADOBJECTMANAGERDATA 63](#_Toc503345436)

[RC-RELOADXREF 63](#_Toc503345437)

[RC-REMOVECONNECTION 63](#_Toc503345438)

[RC-REMOVEPATHS 63](#_Toc503345439)

[RC-REMOVEUNKNOWNDATA 63](#_Toc503345440)

[RC-RESETLAYER 63](#_Toc503345441)

[RC-RESETLUAFORMULAS 63](#_Toc503345442)

[RC-RESETTEXTATTRIBUTES 63](#_Toc503345443)

[RC-REVERSEALIGNMENT 64](#_Toc503345444)

[RC-RUNSCRIPT 64](#_Toc503345445)

[RC-SAVEOBJECTTYPEDEFINITIONSINDRAWING 64](#_Toc503345446)

[RC-SELECTBYHANDLE 65](#_Toc503345447)

[RC-SELECTSIMILAR 65](#_Toc503345448)

[RC-SETALIGNMENTGEOMETRYFROMPOLYLINE 65](#_Toc503345449)

[RC-SETBLOCKCOLOR 65](#_Toc503345450)

[RC-SETLAYERPROPERTIES 65](#_Toc503345451)

[RC-SETLENGTHPROPERTY 65](#_Toc503345452)

[RC-SETOBJECTTYPEDEFINITIONS 66](#_Toc503345453)

[RC-SETPATH 66](#_Toc503345454)

[RC-SETPATHSFROMAREA 66](#_Toc503345455)

[RC-SETTEXTATTRIBUTEORIENTATION 66](#_Toc503345456)

[RC-SHOWALIGNMENTNAME 66](#_Toc503345457)

[RC-SHOWATTACHMENTLINES 66](#_Toc503345458)

[RC-SHOWDISTANCETOALIGNMENT 67](#_Toc503345459)

[RC-SHOWELEVATIONABOVEALIGNMENT 67](#_Toc503345460)

[RC-SHOWFOULINGPOINTS 67](#_Toc503345461)

[RC-SHOWGEOMETRY 67](#_Toc503345462)

[RC-SHOWHELP 67](#_Toc503345463)

[RC-SHOWLAYERS 67](#_Toc503345464)

[RC-SHOWMILEAGE 68](#_Toc503345465)

[RC-SHOWMILEAGECHANGE 68](#_Toc503345466)

[RC-SHOWMODULES 68](#_Toc503345467)

[RC-SHOWPOSITION 68](#_Toc503345468)

[RC-SHOWPOSITIONINALIGNMENT 69](#_Toc503345469)

[RC-SHOWPOSITIONREFERENCEINALIGNMENT 69](#_Toc503345470)

[RC-SHOWPROFILE 69](#_Toc503345471)

[RC-SHOWREFERENCEALIGNMENT 69](#_Toc503345472)

[RC-SHOWREFERENCEMILEAGE 69](#_Toc503345473)

[RC-SHOWRELATIONLINES 69](#_Toc503345474)

[RC-SHOWTOPOLOGY 69](#_Toc503345475)

[RC-SHOWTUTORIALS 70](#_Toc503345476)

[RC-SHOWTVDSECTIONSTRANSIENT 70](#_Toc503345477)

[RC-SHOWTWORAILS 70](#_Toc503345478)

[RC-SHOWVERSION 70](#_Toc503345479)

[RC-SHOWWARNINGINCADTEXTWINDOW 70](#_Toc503345480)

[RC-SNAPTOSELECTEDALIGNMENT 70](#_Toc503345481)

[RC-SWITCHBLOCK 70](#_Toc503345482)

[RC-TESTER-AUDITSWITCHCONFIGURATION 70](#_Toc503345483)

[RC-TESTER-DRAWINFRASTRUCTUREVISUALIZATIONS 70](#_Toc503345484)

[RC-TESTER-EXPORTTORAILML 71](#_Toc503345485)

[RC-UNLOADXREF 71](#_Toc503345486)

[RC-UPDATEAREA 71](#_Toc503345487)

[RC-UPDATEOBJECTTYPEDEFINITIONSWITHMAPPING 71](#_Toc503345488)

[RC-UPDATESINGLELANDXMLALIGNMENT 71](#_Toc503345489)

[RC-WEBPAGESANDFAQ 71](#_Toc503345490)

[RC-XREFSELECTOFF 72](#_Toc503345491)

[RC-XREFSELECTON 72](#_Toc503345492)

# Welcome

**Willkommen! Bienvenue! Welcome! Dobro pozhalovat’! Velkommen!**

Welcome to RailCOMPLETE – the key to modern and efficient modeling of railway infrastructure.

RailCOMPLETE turns your model into an electronic relay stick which enables owners, designers, contractors and testers to organize and edit railway data as well as produce tables, 2D drawings and 3D visualizations for large multi-discipline railway infrastructure projects. RailCOMPLETE offers support during design, ready-for-construction reviews, construction, acceptance testing and as-built documentation. RailCOMPLETE keeps data synchronized and updated and automates the tedious process of producing consistent and error-free tables and drawings. Just move your objects and edit their properties as needed, click the Refresh button and export your model with updated and consistent tables, drawings and 3D visualization.

## RailCOMPLETE is a CAD system plug-in

RailCOMPLETE operates as a plug-in to your CAD system, which means that you must have access to a compatible CAD system running on a reasonably powerful computer. Currently, RailCOMPLETE will run under the three most recent editions of AutoCAD®, either plain AutoCAD or one of the many add-on editions such as Map 3D, Civil 3D etc. RailCOMPLETE may not work fully with ‘Lite’ versions (LT), ‘student edition’ versions (SE) or free trial versions of the CAD system.

## RailCOMPLETE is a professional and versatile tool

RailCOMPLETE has been developed in order to provide professionals within the railway infrastructure community with a versatile and modern tool for entering, visualizing, editing, checking and exporting information. As the usage of RailCOMPLETE expands, all involved parties will enjoy the freedom and the ease of sharing complex and detailed information in projects and in operations.

## RailCOMPLETE is based on a safe data format

In addition to storing drawing details in the DWG format, RailCOMPLETE fully exploits the meta-data storage features of AutoCAD. This means that data such as model names, article codes, relations between different objects, construction stage information etc is stored in a de facto standard AutoCAD format. RailCOMPLETE has been designed to protect internal data from inadvertent tampering. In addition, there are many ways to audit the integrity of a RailCOMPLETE model and to repair damaged data.

## RailCOMPLETE protects your engineering investment

RailCOMPLETE has been designed to withstand the cruel effects of time. As time passes by, software and file formats will eventually be replaced with newer versions. Customers change the way they represent their assets, spare parts go out of production and new asset variants appear. The AutoCAD suite of tools and file formats have been backwards compatible since the 1990’s, in the sense that newer versions can always read older files and convert them to a more recent file format. Commands and internal computer interfaces are seldom deprecated, only expanded. Since a RailCOMPLETE computer file is a plain AutoCAD DWG file, we believe that it will continue to be readable also in the decades to come.

## RailCOMPLETE genetics – Object Type Definitions

The challenge of changing assets over time has been met in RailCOMPLETE by mimicking the powerful concepts found in genetics. Picture a railway project as a living creature, and a single computer file as one of its many cells. RailCOMPLETE models this creature’s tiniest parts using a kind of ‘genetic code’, which we call the Object Type Definitions (or ‘OTD’ for short). The objects that are inserted into the computer model are all described in this OTD, and the whole OTD is stored in every RailCOMPLETE computer file, just as the whole DNA code will be found in every cell of a body. In a healthy normal individual – i.e., in a well-managed railway project – all computer files share the same OTD, just as all cells in a body share the same DNA. If a computer file is introduced with different but still compatible OTD, then minor issues may occur, similar to an immune system response in a living individual.

In order to upgrade an existing object’s model files to a more recent version of the OTD, RailCOMPLETE provides a mapping mechanism, which involves replacement of RailCOMPLETE system files and mapping of deprecated representations into new representations.

## RailCOMPLETE Agents

The people making changes to an existing OTD – the genetic mutations, if you like – are so-called ‘RailCOMPLETE Agents’. A RailCOMPLETE agent is a trusted company which knows your local railway administration’s needs in detail, and which also knows the inner workings of the RailCOMPLETE software. Whenever a new object needs modeling, or an existing one needs to be adjusted, then the agent will provide this service on behalf of the local railway administration. The agent may be part of the administration itself, or it may be a competent local engineering company. The agent will make the new OTD accessible for download to all relevant registered users working for the railway administration, and local superusers will guide their colleagues in the OTD update process.

## RailCOMPLETE Patches

Although we should avoid it, from time to time there will be a need for changing an existing function’s inner way of working such that the model data for a railway project must be modified in ways that the ordinary OTD update process cannot handle. In such cases the RailCOMPLETE development team provides a so-called ‘patch’ on the download site. This patch is simply yet another RailCOMPLETE command that you download and install on your computer, and which will be automatically run when needed – the patch figures out this by itself.

## The RailCOMPLETE Business Case

Using RailCOMPLETE for your projects over time will boost the quality of the following processes:

* Data referencing between different engineering entities in your project
* Data sharing between different companies, or between consultant, customer and operations
* Data format longevity
* Re-use of data – unpack your old projects and find model data there in a still readable format
* Re-use of data – each link in the chain from conception to operations adds to the existing model
* Increased data quality – each link checks and refines data from its predecessor
* Cross-discipline integrity checking of the model – everyone works in one connected model
* Premium quality on-site construction drawings and data sheets
* Premium quality as-built data base contents

**FITS ANY RAILWAY**

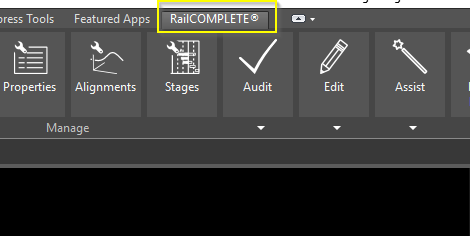
**MODELS EVERY OBJECT**

**CONNECTS EVERYONE**

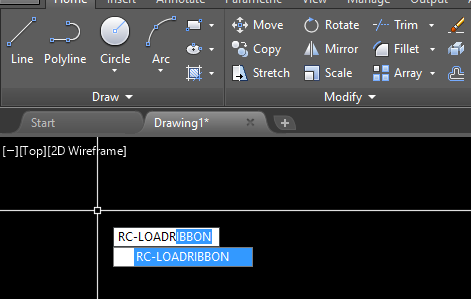
# Getting started

## Starting RailCOMPLETE

Normally, RailCOMPLETE is launched when AutoCAD is started. The RailCOMPLETE buttons should be visible under the RailCOMPLETE pane.



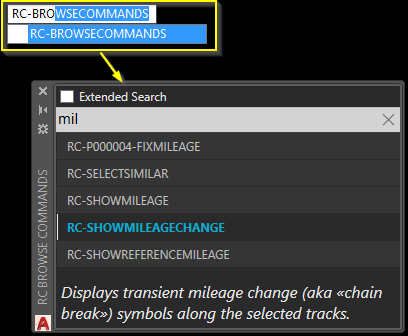
If RailCOMPLETE is not visible inside AutoCAD, the menu can be loaded with the RC-LOADRIBBON command.



## RailCOMPLETE commands

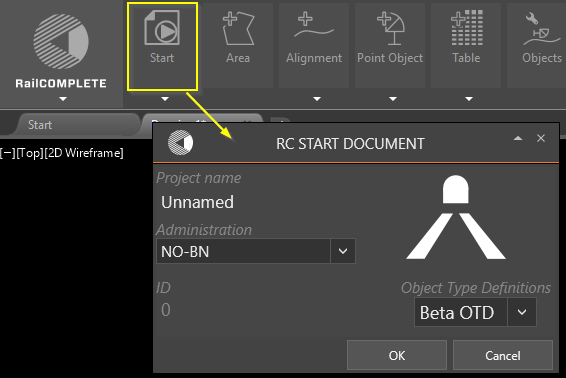
RailCOMPLETE have more than 100 built-in commands. All RailCOMPLETE commands starts with a “RC-” prefix. In general, a command name consists of an action and an object. Typical examples are RC-MOVEATTRIBUTES, RC-CREATEAREA and RC-SHOWGEOMETRY.

To get a list of all available RailCOMPLETE commands, start the command RC-BROWSECOMMANDS, as shown below.



## Start a new RailCOMPLETE document

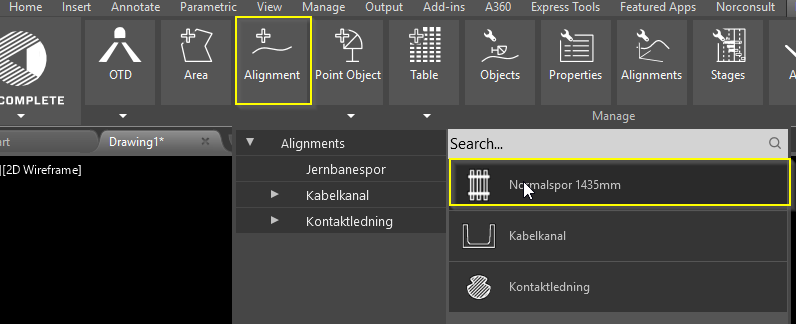
To start a new RailCOMPLETE document, open an existing or a blank AutoCAD document. In the ribbon, select *Start* New Document and give the project a name (optional)

.

When the AutoCAD document has been converted to a RailCOMPLETE drawing, the rest of the icons in the ribbon is enabled, and the *Start New Document* button is changed to *Document Properties*.

## A quick example

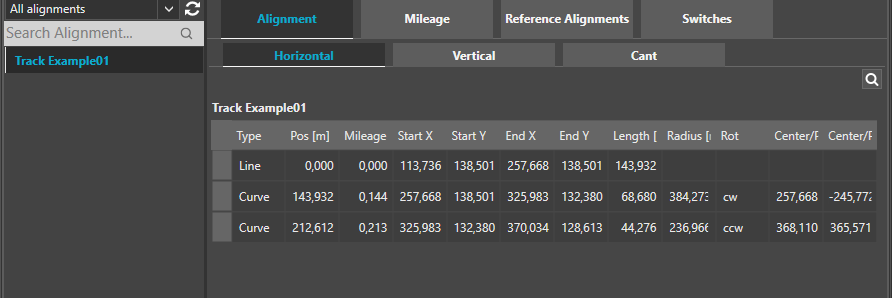
Start a new RailCOMPLETE document, as described above, and create a track by drawing a polyline from the *Create Alignment* button.



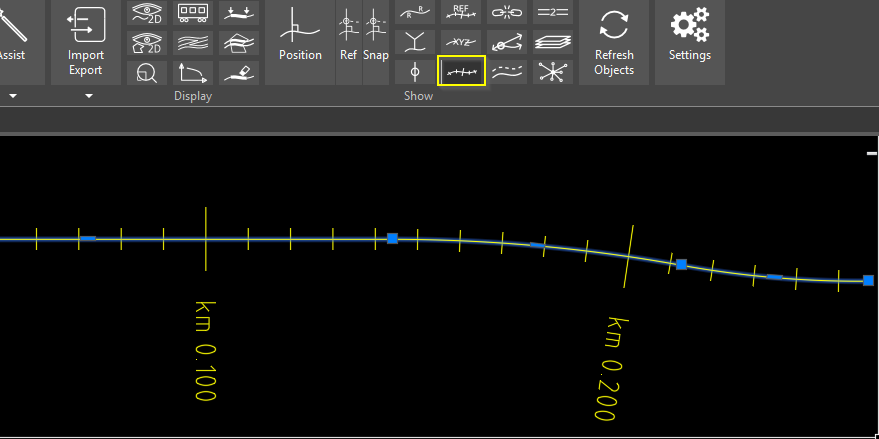
RailCOMPLETE will ask for an alignment name after the track is created. In this case, we will name the track "Track Example01"



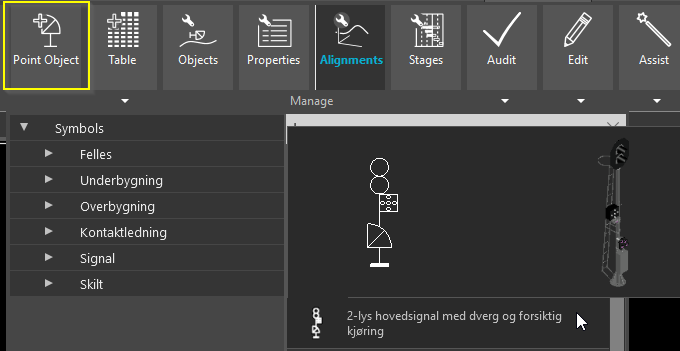
This polyline is now a RailCOMPLETE track, and can be manipulated in RC-MANAGEALIGNMENTS.



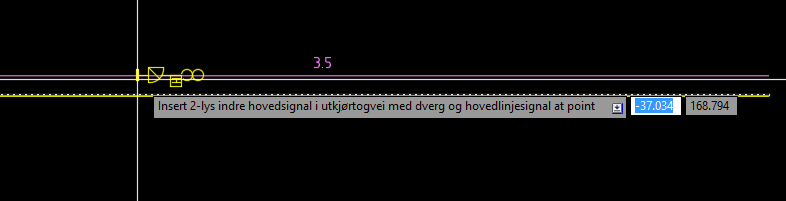
Alignment mileage is shown when the 'RC-SHOWMILEAGE' is selected.



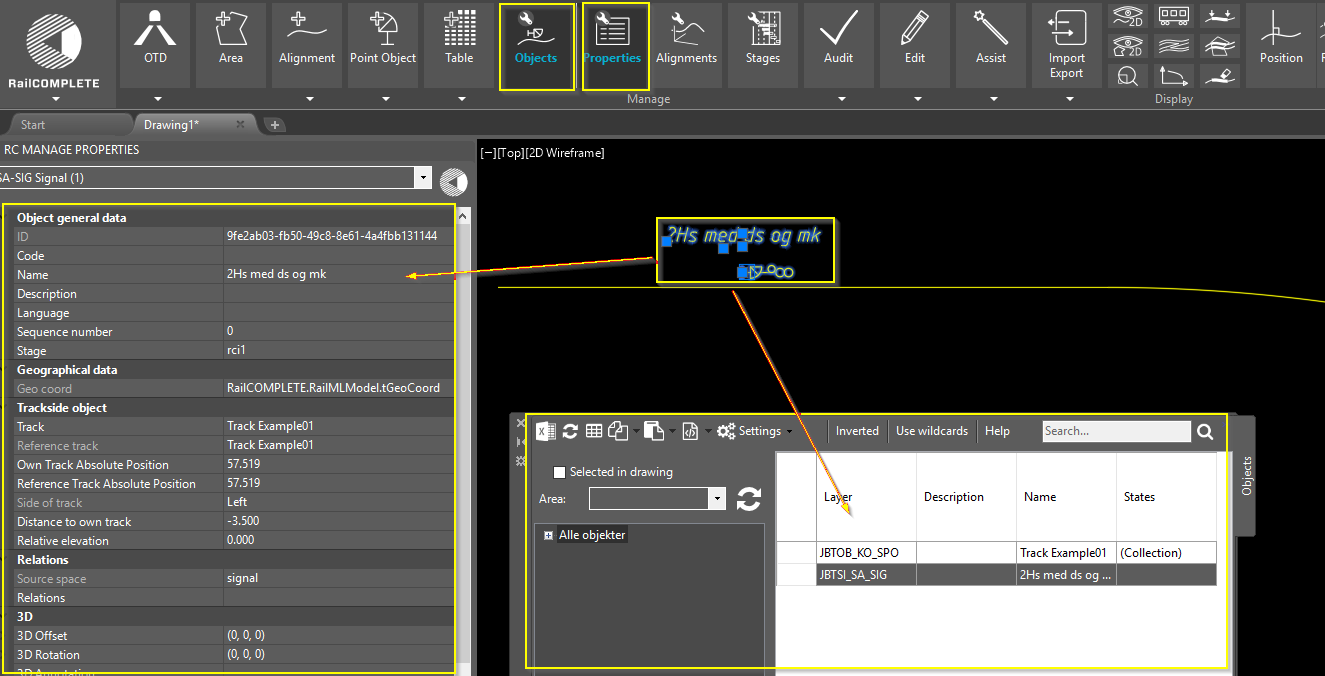
To add objects, click on the Point Object button and navigate to any item. It is also possible to search for objects in the search-field.



Select the object to be added, and snap the item to some place on the alignment.



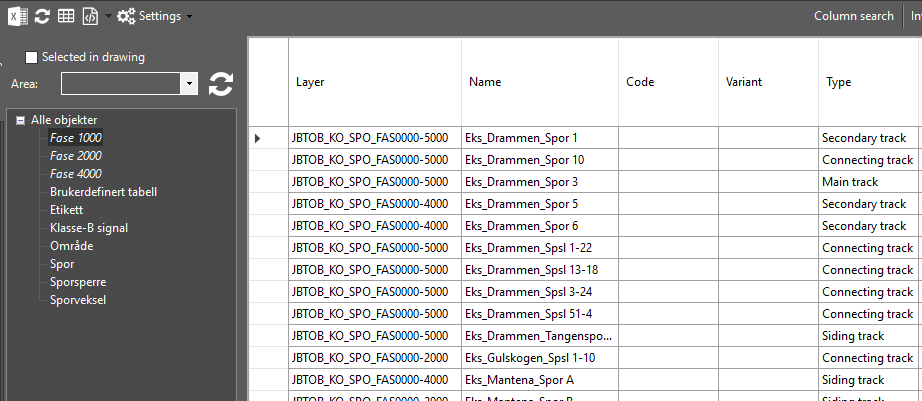
The item has now been created, and it is possible to view and manipulate the object details by opening the Properties window.



# Main features

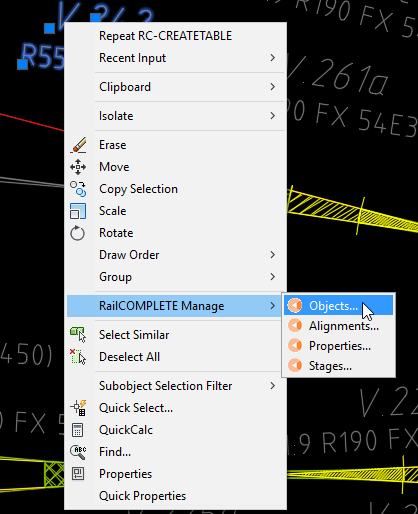
## Manage Objects

All RailCOMPLETE items in the drawing are shown in the Object Manager. This includes all railway objects as well as stages, tables and areas.



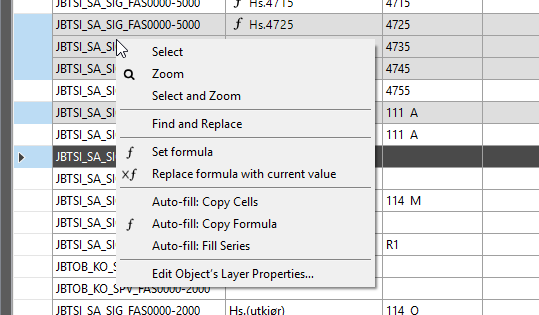
### Locate in Manage Objects

A RailCOMPLETE item can be located in the Object Manager by right-clicking on the item in the drawing, and selecting "RailCOMPLETE Manage -> Object..." as shown below.



### Locate in drawing

When one or more objects are selected in the Object Manager, a right-click, will give the options select and/or zoom.



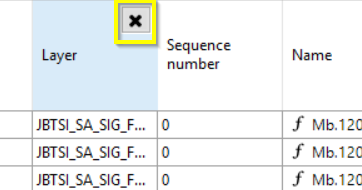
### Add and remove columns

To get all available columns, right-click on the column header, and select Add.

A screenshot of a cell phone

Description generated with very high confidence

Click the X on the column header to remove the column.



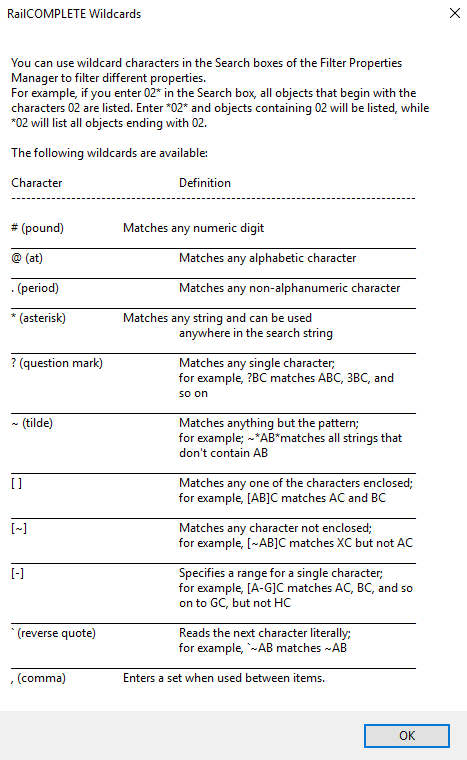
### Searching and filtering

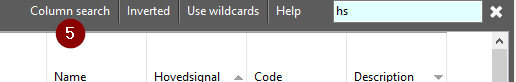
The Object Manager has powerful search and filter mechanisms. Filtering can be done by selecting the desired object type or types in the left side browser (1) or by entering a search pattern in the search field (2). Only filtered rows and unhidden columns are searched. The categories shown in the object type browser may also be custom object type filters – right-click an existing item to add, edit or delete an object type filter.

A screenshot of a computer

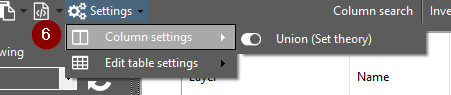
Description generated with very high confidence

Select between searching using wildcards or just regular text (3). The AutoCAD wildcard syntax applies, press Help (4) to see details.



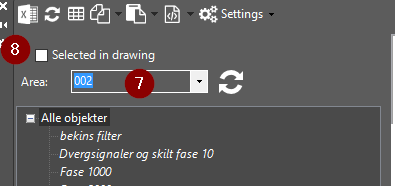


Yet another search method is to use Column search (5), where each column gets its own private search pattern to be used in that column only. Under Object Manager Settings (6) you can select between Union or Intersection search modes, i.e. picking Object Manager rows that match at least one column search pattern, or only those that match all the column search patterns.



All of the search methods mentioned above may be combined in order to narrow down your row selection (objects).

To further narrowing down your search, you may select a named area in the drawing (7), i.e. considering only objects found inside that area.

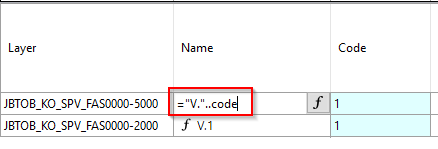


Another type of search mechanism is to select objects in the CAD system modelspace using the CAD system’s Quick Select or pick-them-yourself methods, and checking the “Selected in drawing” box (8). in the Object Manager.

### Using formula

The fields (the cells) in the Object Manager display the current values of their object’s properties. Such properties may either hold a direct value, or a formula which evaluates to a derived value. E.g., a ‘name’ property can be derived from a static text “V.” concatenated with the ‘code’ value from that same object, giving the formula ‘”V.”..code’. The ‘=’ equality sign is a convenient way to tell the Object Manger that you are now entering a formula and not a direct value.

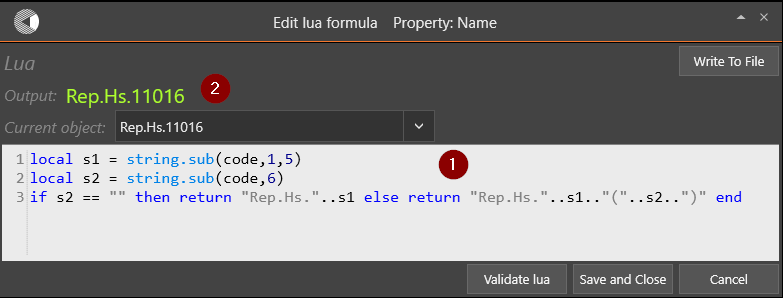
We tend to use the ‘name’ property for object names to be shown in drawings and tables, and the ‘code’ property for a terser version. You will also often see that the ‘code’ property is formulated by involving related object’s codes, or formulated using the object’s own ‘sequence number’ property, e.g. numbering outgoing turnouts’ ‘code’ property as ‘(seq\*2)+1’.



To remove a formula and replace it with its current value, just erase the formula, or right-click and select “Replace formula with current value”.

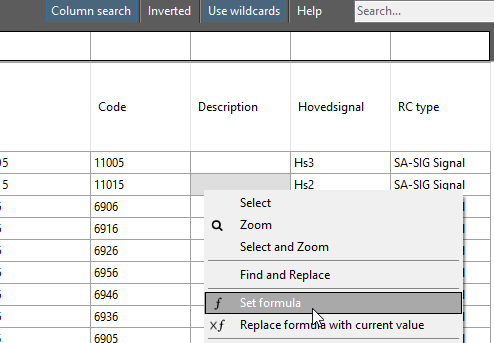
To start the formula editor, click the ‘f’ icon in the cell, or right-click and select “Set formula”.

Use the formula editor (1) until you get the desired output (2). The output field will show “Null or syntax error” if the formula contains errors. Click the Validate button to see where your problem is.

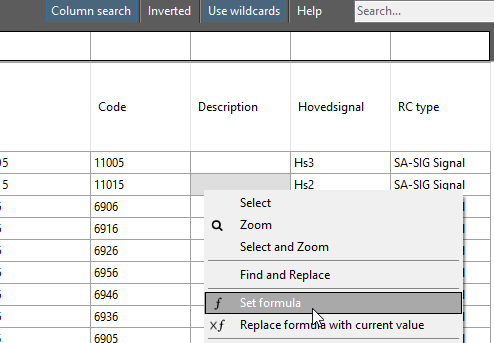


### Set formula

The Object Manager also contain a *script window* for testing out formulas. Right-click inside a cell, and select Set Formula.



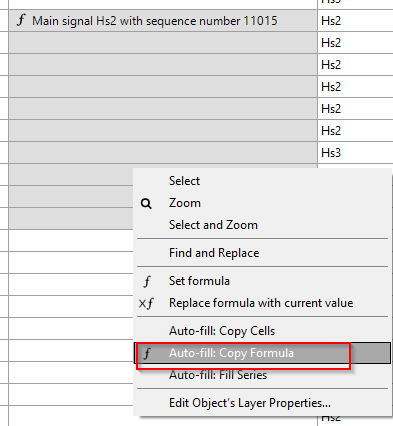
In this example, we will give the *description* field "Main signal with sequence number ". This can be tested in the Set lua formula window.



Use the text editor (1) to get the desired output (2). The output-field will show "Null or syntax error" if the formula contains errors.

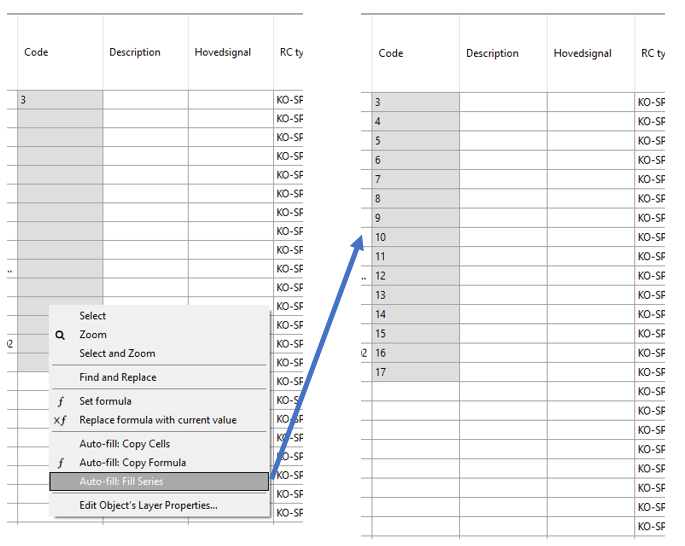
### Auto-fill: Copy formula

To copy formula, use the Auto-fill: Copy formula. Be sure to have one cell with formula within the selection.



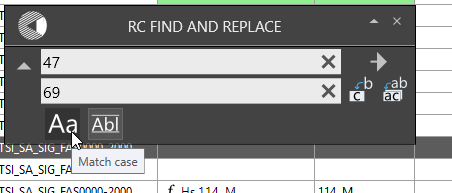
### Auto-fill: Fill series

The Object Manager offers a smart way to fill series. By selecting a set of fields and select the *Auto-fill: Fill Series*, the empty fields will be filled with a sequence number, as shown below.



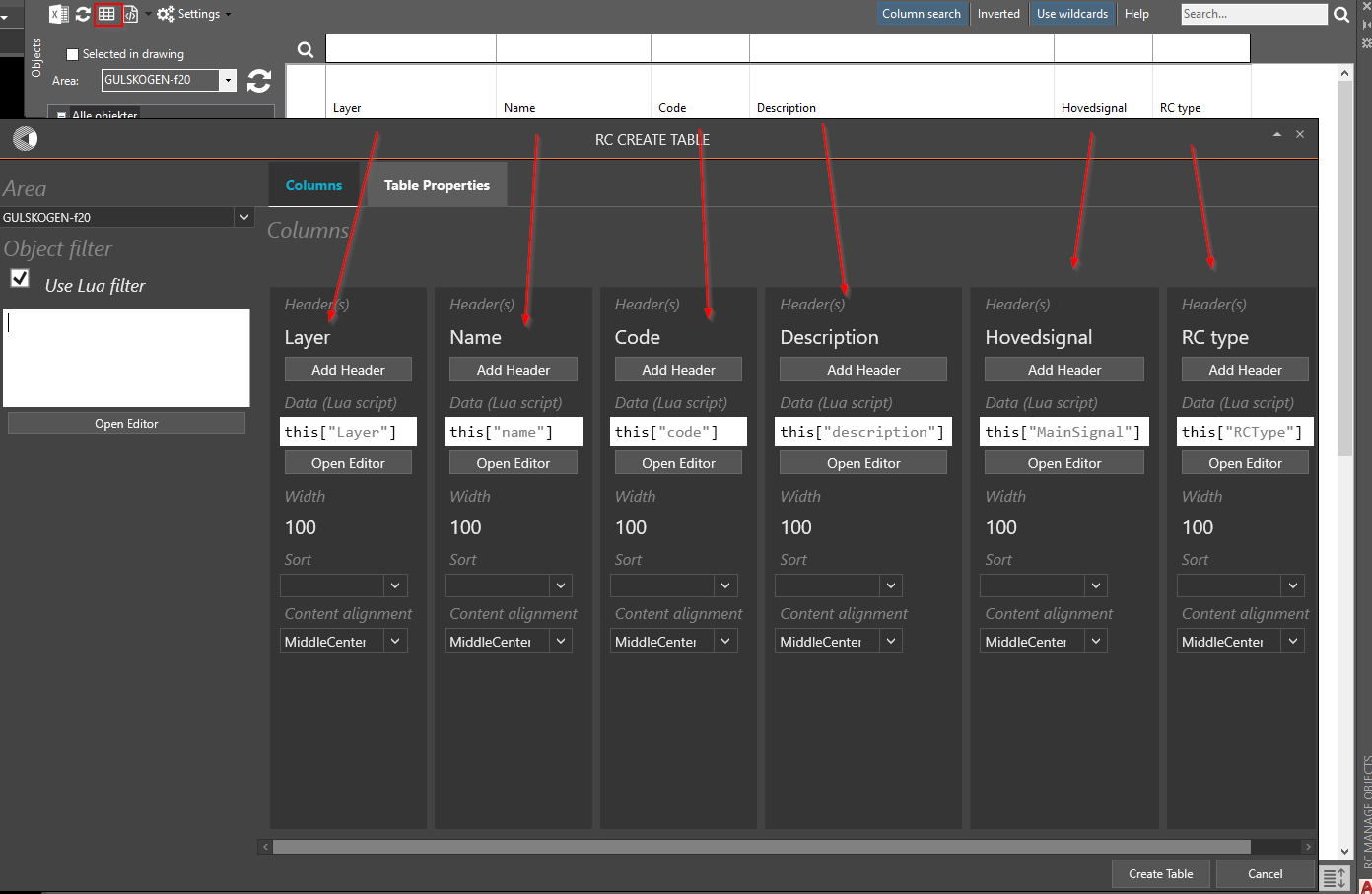
### Find and replace

Use the Find and replace (right-click inside the Object Manager) to do mass-update of values.



### Create table from Object Manager

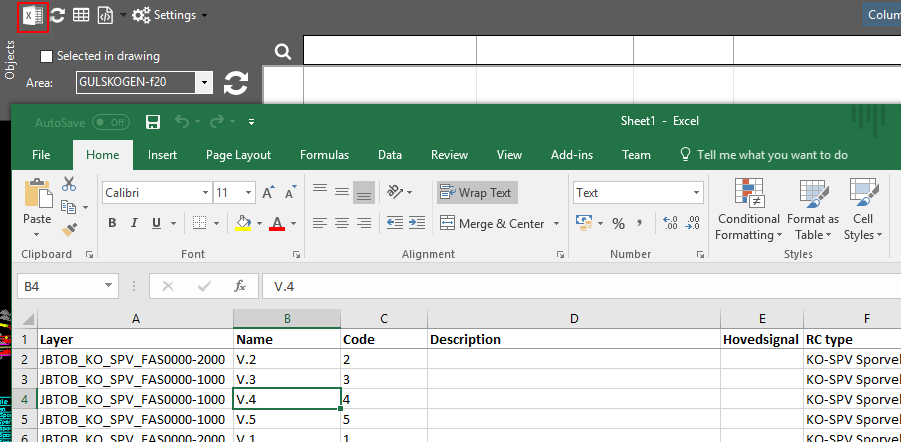
A quick way to create a table inside the drawing, is to click the "Create Table" button at the top right corner in the Manage Objects window.



### Export to Excel

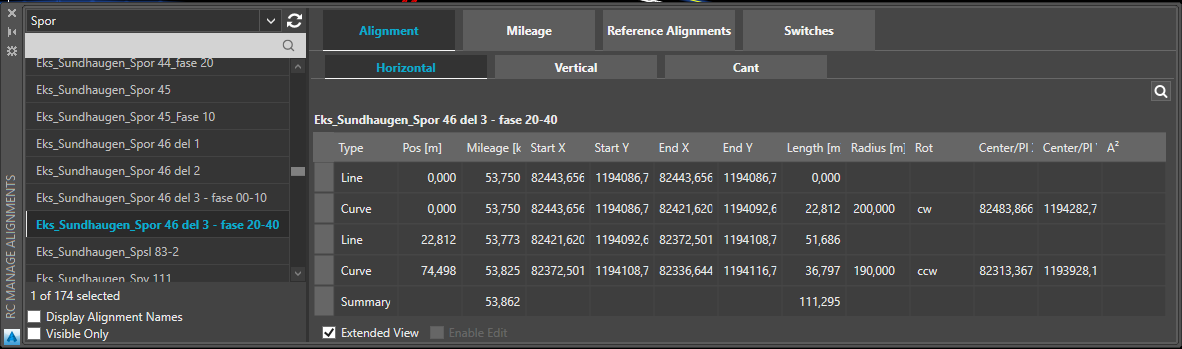
Clicking the Export to Excel button in the top left corner will open an Excel spreadsheet with the same contents.

Note that search patterns and filters which are active in the Object Manager will also be applied during Excel export.



### Manage Alignments

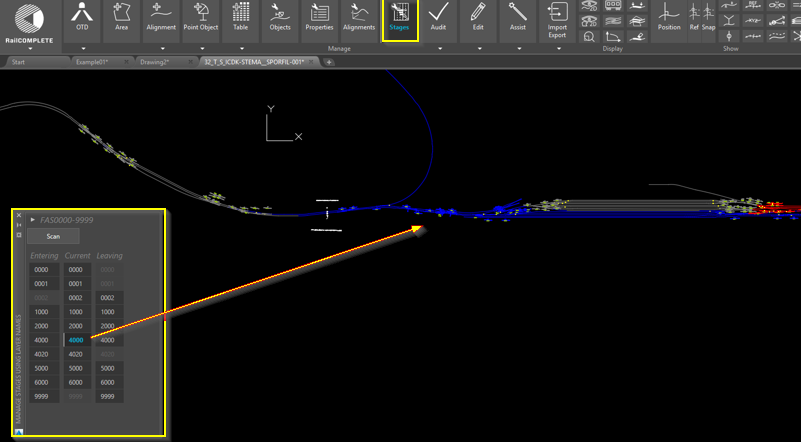
All RailCOMPLETE alignment items are shown in the Manage Alignments window.



The Alignment Manager offers the possibility to...

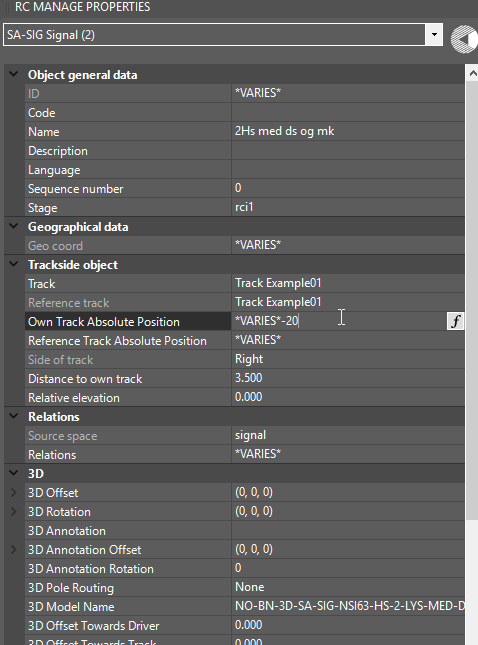
* *View alignment horizontal, vertical and cant data.*
* Manage mileage data and mileage inheritance.
* *Manage reference alignments.*
* View switches.

## Manage Stages

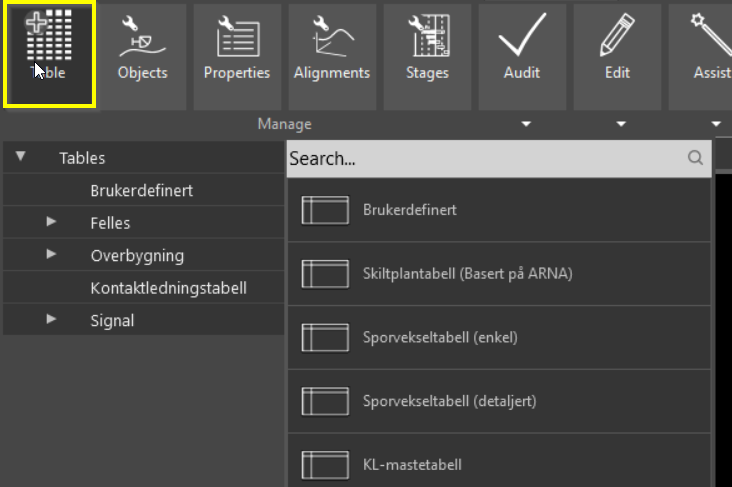
The Stage Manager scans the CAD system layers for information about the railway model’s construction stages. The scanning is guided by a user defined Stage Pattern. By selecting one or more items from the Entering, Current and / or Leaving stage columns, you can see what is entering at the beginning of a phase, what is present during that phase’s duration, and what is leaving towards a phase. E.g., if stage info such as “\_Phase 20.00 – 40.00” is found inside a CAD layer name, then for instance selecting “Entering 20.00”, or “Current 30.00” or “Leaving 40.00” will turn that layer on and make its CAD objects visible. In order to see time-critical construction conflicts, simply select Entering xx.yy and the Leaving xx.yy, where xx.yy denotes the start of a particular operational stage. 

## Manage Properties

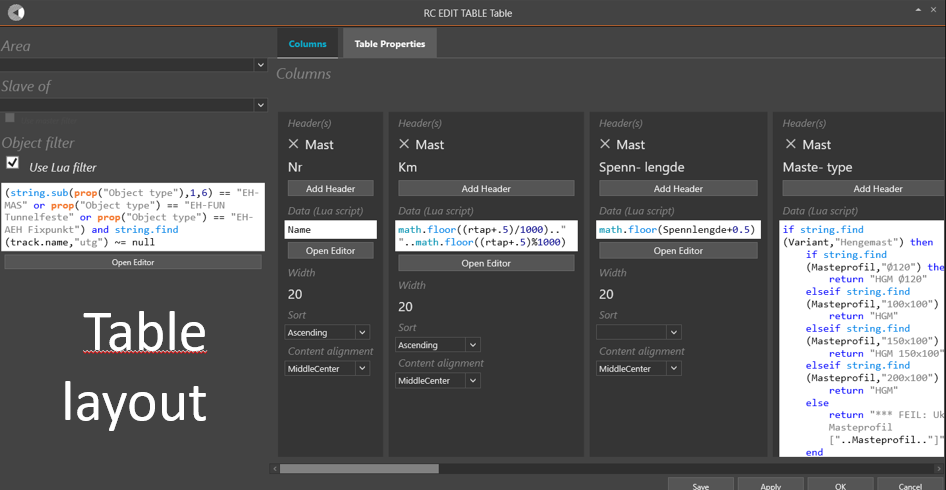
All railway properties are displayed in the Manage Properties window (as well as the Manage Objects window). Use this window to manipulate one or more railway objects.



## Manage Tables

Customized and predefined tables can be created from the Create (+) Table button in the ribbon, or from “Create Table” in the Object Manager. 

A column’s content is defined by a formula. A formula may actually be a complex and large program which returns exactly one value. The language used for such formulas / programs is Lua. Please refer to the open literature for details on Lua.

Formatting and sorting may be specified for each table column.

# Q & A

## Speed issues

*Q:* Updating a lot of objects simultaneously is time consuming.

*A:* When doing mass update, AutoCAD can sometimes use a lot of "overhead" during the update. By setting the system variable GRIPOBJLIMIT to i.e. 10, AutoCAD will not do this drawing operation while objects are updated.

Setting the GRIPOBJLIMIT to 10

## Manage Stages

*Q:* I am using the Stage Manager in V1.0 (RC-ManageStagesUsingLayerNames), but some of my layers won't show up.

*A:* There is a bug in AutoCAD which quite often causes random layers to remain frozen even if they are turned on by the RailCOMPLETE software. This bug only occurs to layers that reside in XREF’ed drawings. To fix it, do the following: Open the AutoCAD layer manager with command LAYER and find a layer that is currently visible, i.e. the light bulb is bright (ON) and the sun is shining (THAWED). Just click on the sun icon and click again on the snow crystal that shows up to get the sun back again. Your invisible layers are now turned on and are consistent with the current Stage Manager layer selection again.

*Q:* I am using the Stage Manager in V1.0 (RC-ManageStagesUsingLayerNames). When I select a certain stage as current stage, there are still partial graphical elements which are not made invisible, but I know for sure that these elements belong to objects that are not present in that operational stage.

*A:* A possible reason for this behaviour can be that you have set as current layer (with the AutoCAD LAYER command) the layer that the object(s) in question reside on. When the Stage Manager tries to turn off that layer, it is not allowed to do so due to the way AutoCAD operates – you can’t Freeze the current layer, i.e. it remains Thawed. But the Stage Manager is allowed to turn Off that layer (the light bulb in AutoCAD layer manager goes out), so all graphical elements residing on this Thawed object layer will become invisible. However, if the object – for instance a switch – has graphical elements residing on other layers, say a layer for track center lines, then these parts of the switch will be visible as long as that layer is Thawed and On. The solution is to change the current layer to for instance layer 0 with the AutoCAD layer manager, just double-click on layer 0.

## Object Type Definitions

Q: I made an update to the OTD in my model file, and now I cannot read positions any more. I cannot select objects residing in XREFs either.

A: The present V1.0 requires that the OTD in XREFs must be the same as the OTD in the model file you are looking at.

## Position

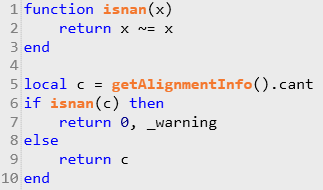
Q: I turn on the Position tool, but the transient position info doesn't show in my drawing's modelspace.

A: You may have activated the Position Snap mode. If the Position tool has been de-activated and is then re-activated, then you must select a track explicitly again in order to turn on the position. Otherwise it would not know which track to calculate positions relative to.

## LUA

Q: How can I check if an alignment has cant information (or gradient) or not?

A: When no information is available, Lua may return a “NaN” value. This is a very specialized value, which is the only one that is not equal to itself (!), that is, (NaN ~= NaN) is actually True. See the following example – a local function isnan(x) and an example where cant or 0 (and a warning triangle appears) is extracted from an object’s own alignment:



## RailCOMPLETE dialog window is not visible

Q: I have started command RC-AutoCreateSwitchesAndConnections, but the dialog window does not appear on my computer screen(s). When I click on the AutoCAD-icon at the bottom Windows line of my computer screen, I can see a thumbnail picture of my window, so I know it’s there somewhere, outside my screens.

A: This is a common Windows problem which typically occur when you undock your laptop from a docking station and either re-dock it in the same or in another docking station. Windows gets confused about where to display new windows.

A workaround is to click on the AutoCAD icon in the bottom Window line on your main computer screen. You shall see a thumbnail version of the missing dialog window. Right-click on the window and select “Move”. Then use your keyboard arrow keys (left, right) until the missing window appears on one of your active screens.

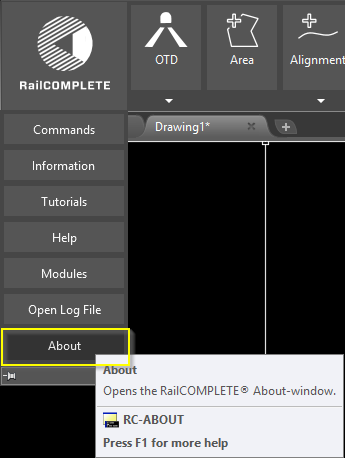
# RailCOMPLETE commands

In this section, all available RailCOMPLETE commands with description are listed.

## RC-ABOUT

**Opens the RailCOMPLETE® About window.**

The About window displays version number, license information as well as the end user license agreement (EULA).



## RC-APPROXIMATEPOLYLINEBYARCS

**Provides an interface for turning the CAD system layers on and off in order to display stage dependent drawing details for objects that reside on the affected layers.**

Objects may be RailCOMPLETE objects or any type of CAD object (circle, line, block, leader, text, XREF etc). Specify the stage pattern as either a - stage pattern, or as a stage pattern, with user defined prefix, entering/leaving infix and suffix patterns, number of digits used for denoting main and sub stage numbers, as well as a user defined main/sub stage separator. For instance, the pattern “MyPrefix”, “2”, “.”, “2”, “-”, “MySuffix” refers to all layers whose names contain “MyPrefix##.##-##.##MySuffix”, where # denotes any single digit 0-9.

## RC-ASSISTCREATESWITCHESANDCONNECTIONS

**Evaluates and fixes switch objects and connections between RailCOMPLETE tracks.**

## RC-ASSISTSETMILEAGEPARENTS

**Automatically assigns mileage parent for each alignment in the selection set and derives the mileage from the parent.**

The algorithm will search the database for candidate alignments. The candidate which is closest to the

selected alignment's endpoints will be taken as the mileage parent. Mileage inheritance through the

start point takes precedence over the end point when candidates are present at both extremities. An

alignment can derive its mileage from another alignment by assigning a "mileage parent". The alignment

attempts to re-derive its mileage whenever the mileage parent is updated to ensure data consistency.

The mileage inheritance can also be edited manually in the alignment manager.

## RC-ATTACHXREF

**Attach RailCOMPLETE documents to drawing as Xrefs. A RCXref is created for each attached document.**

## RC-ATTSYNC

**Applies attribute changes in a block definition to some or all block references. Preserves attribute geometry.**

## RC-AUDITALIGNMENT

**Audit alignment geometry, data consistency, search for missing data, etc.**

Some issues are also fixed during audit.

## RC-AUDITALL

**Evaluates the integrity of RailCOMPLETE® objects in current drawing and fixes possible errors.**

## RC-AUDITALLRELATIONS

**Audit relations in drawing and re-connects one-way relations if possible.**

## RC-AUDITUNIQUEIDS

**Assigns new id to objects with duplicate id's**

## RC-BINDXREF

**Converts Xref into a block, making it a permanent part of the drawing.**

## RC-BREAKALIGNMENTS

**Break one or more alignments. Erases the old alignment and transfers all elements and attached objects to the new alignments.**

## RC-BROWSECOMMANDS

**The command browser is a convenient way to search through all available commands.**

Use extended search (1) to search through the command description.

Search for the command or description in the search field (2).

Use arrow keys to step up/down.

Start the command by double-click or by hitting enter on the current command (3).

The command description is shown below (4).

## RC-CALCULATEFRESNEL

**Calculates fresnel integrals SS = \int\_0^x sin(t^2) dt, CC = \int\_0^x cos(t^2) dt.**

## RC-COPYANNOTATIONSTODRAWING

**Converts transient annotations to permanent objects in drawing.**

## RC-COPYBRAKINGCURVETOCLIPBOARD

**Copies the last executed braking curve calculation to clipboard.**

## RC-COPYPOSITIONTOCLIPBOARD

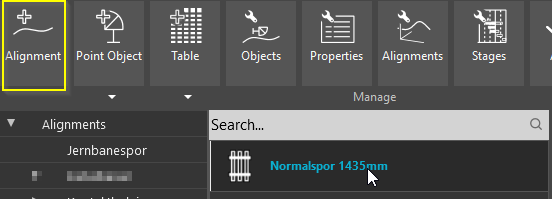
**Copies the currently displayed position-details to the clipboard.**

## RC-COPYSELECTIONALONGPATH

**Produces copies of the selected group of signals at the desired increments (which may be zero). Multiple copies can be made by adjusting the Amount variable. The additional copies will be placed an increment's length away from the previous copy, resulting in an equidistant sequence of copies of an arbitrary size decided by the user.**

## RC-CREATEALIGNMENT

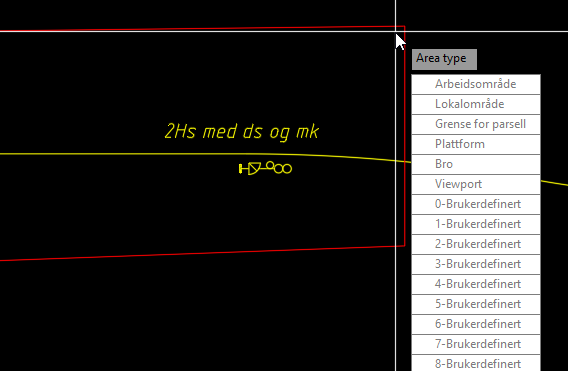
**Creates a railway alignment by specifying points or by using an existing polyline(s), both 2D and 3D polylines will work.**



Choose between track, duct, or catenary alignments etc. V1.0 can only create track objects.

## RC-CREATEAREA

**Creates an RC Area by specifying points or by using an existing polyline.**



An Area type is specified after the area boundary is defined.

## RC-CREATECONNECTION

**Creates a connection between two adjacent alignments (e.g., two tracks without joining them), or between an alignment and a branching alignment (e.g., a switch). See also RC-JOINALIGNMENTS. Does not work for objects in XREFs.**

## RC-CREATELAYERMAPPING

**Create a layer mapping file for 3D export.**

## RC-CREATEOBJECTTYPEDEFINITIONSMAPPING

**Compares the contents of the selected Object Type Definitions (OTD) file with the current OTD in computer memory, and lets the user decide how to translate old OTD data into their new property names and enum values.**

A mapping tool opens and displays a color-coded view of all OTD differences.

Select an object type from the left pane and select a red or yellow property to map in the right pane. Green properties have no change and should not be touched. New properties with no precursor shall be tagged “do not map”.

Properties that have merely changed name shall be mapped by selecting which new property has taken over the old property’s role.

If the attributes of an existing property are of enumeration type, and if the enum values have changed, then you must map enum by enum value for that property.

## RC-CREATEPOINTOBJECT

**Creates a new RailCOMPLETE object.**

The new object tries to attach itself to a nearby alignment, most often with a certain snap distance. Snapping can be toggled during insertion, and may also be overridden by moving the object after insertion.

### ALIGNMENT AND POSITION

The alignment that a point object has attached itself to is known as the object’s “own alignment”. The object’s mileage is defined at insertion by its own alignment’s name and the resulting own alignment absolute position, measured at a straight angle from own alignment to the object’s insertion point in the XY plane. There is a unique correspondence between the object’s mileage together with its perpendicular distance from own alignment, and the object’s position in the XY-plane. The object’s Z-coordinate is made up of the object’s elevation relative to the local alignment elevation, and the own alignment’s actual Z-coordinate – i.e., the alignment’s elevation above sea level – at that mileage. The RailCOMPLETE Properties tool will show these position values at any time, as well as many other derived values, and lets you modify them through adding to, subtracting from or replacing the numerical values for mileage, reference mileage, distance from alignment etc.

### COORDINATE SYSTEM

The end user of the entire model, at construction time, will interpret XY positions as points according to a certain mapping of the earth, for instance UTM-32. This mapping must be the same for all the drawings constituting a model. Likewise, the Z coordinate will be interpreted as an elevation according to a particular model of the earth’s surface.

### REFERENCE ALIGNMENT

Nearby alignments often belong to a named alignment group, constituting a railway line. One of these alignments will usually be nominated as the “reference alignment” for the line as a whole. Each alignment has knowledge of its own reference alignment’s name at every alignment position. Thus, a point object has an alternative mileage definition, expressed as the unique reference alignment name and the reference alignment absolute position (rtap), measured at a straight angle from reference alignment to the object’s insertion point in the XY plane.

### DIRECTION

In addition to a definite XYZ position in 3-dimensional space, a point object also has a rotation in the XY-plane. The Z-axis for a point object is always upright. Most point objects will automatically orient themselves along their own alignment. Most objects must be oriented up or down along the alignment, for instance optical signals. The direction of intended use for a point object can be changed during the insertion process, and later. Some point objects, such as cabinets, have four natural orientations, ‘up’ meaning that a driver running in the alignment’s ‘up’ direction will see the front door of the cabinet, and ‘down’ meaning that he will see the cabinet’s rear side. With a cabinet oriented in direction ‘both’, train drivers in both running directions will see the cabinet door, opening towards the alignment. With direction ‘none’, no driver will see the cabinet’s door since it opens away from the alignment.

### OBJECT PROPERTIES

The object’s data fields are populated with initial values as defined by the corresponding definitions in the relevant administration’s Object Type Definitions (OTD) file, bundled with your RailCOMPLETE installation. Data values may alternatively be defined with an individual script, which means that their definite values will be the result of a runtime evaluation and will be based on the object’s position and its relations to other objects. Such values defined by formulas are refreshed both at creation and later when the object is moved or otherwise changed, or when its related objects change. However, to be 100% certain, please refresh your objects explicitly before you export them to 3D, use them in a table etc.

### POSITION IN DRAWING VS. “REAL” XY POSITION XY

The point object’s 2D symbol may be moved to an offset position from its real position in the XY-plane using the square grip, in order to tidy up a messy 2D drawing with overlapping symbols. A transient dashed line will indicate for you where the real XY position is at any time when you move the object around.

### ATTACHING OBJECTS TO EACH OTHER

A point object may be attached to another point object. When selecting an object with an attachment, a transient arrow will indicate the direction of attachment. Such an object may inherit its “real” position from the parent object. A signal board will by default inherit its real XYZ position from a nearby foundation, to whom it is attached to as part of the point object creation dialog. This allows for a board tail to be drawn from the board’s 2D symbol to its foundation, making it graphically clear which running direction and side of alignment the board is meant for. Poles and signs are likewise treated. Boards can be stacked on the same pole by attaching the next board to the previous board. In such a case, the attachment tail is suppressed for the stacked board, and its real XYZ position is derived from the foundation at the bottom of the stack. The position inheritance can be modified at any time for any point object.

### TEXT ATTRIBUTES

Many point objects need to be shown in 2D drawings with text attributes. These text attributes are defined in the OTD. Text attribute contents are driven by a dedicated data item found in the base class for the point object in question, or a custom attribute, as detailed in the OTD. There are commands available for modifying or resetting such text attributes, for instance rotating texts along a alignment or along a paper drawing’s borders.

### 3D VISUALIZATION

Most point objects have an associated 3D model that can be generated using the Export3D command. Using the Properties or Object Manager tools, the 3D model can be offset from the point object’s insertion point with the 3D Offsets X/Y/Z, as well as rotated in space with the 3D Pitch/Roll/Yaw data items, accessible both in the Properties and the Object Manager tools. This allows for instance a wayside board to be drawn vertically above its foundation, or sideways towards the alignment when being wall-mounted in a tunnel. To allow for quick 3D modeling of poles and masts, every point object can use horizontal or vertical pole routing at 3D export time. Please refer to the Export3D command reference documentation.

## RC-CREATETABLE

**Creates a new RailCOMPLETE table.**

Create a variety customized tables by selecting objects and parameters. Values can also be derived from other objects by using LUA script.

## RC-DETACHXREF

**Detach Xrefs from RailCOMPLETE document.**

## RC-DISPLAYBRAKINGCURVEEBICAB700

**Prompts the user for a path and for Ebicab700 ATC braking curve parameters in SI units. Displays braking curve information, which may be copied to the drawing.**

Moving the CAD cursor along the path shows instantaneous values for time, deceleration, speed and distance (does not apply to V1.0). Documentation of the braking curve parameter is echoed to the CAD system’s text window.

## RC-DISPLAYSIGHTAREA

**Displays the area, along the alignment towards one or more objects, that must be free from major obstacles in order to guarantee free sight for the locomotive driver from any distance up to the sighting distance requirement.**

The sighting distance is given by the @Sight property in the object (meters), or the user is prompted for a sighting distance requirement if the object type does not possess the @Sight property.

If multiple paths of the required length exist towards the object, then the combined sight area for all of these is shown. The presence of a 3D model for the object may influence the sight area calculation, see 3D models.

## RC-DISPLAYSIGHTLINE

**Displays a line of sight from an observer located in the middle of a track towards one or more objects.**

The sighting distance in 2D is calculated disregarding the Z coordinate. The sighting distance is specified in the object’s @Sight property in meters. If the object does not possess the @Sight property, or if @Sight is zero, then the user will be prompted for a sighting distance. Sight lines may be computed for multiple objects, where the user-entered sighting distance requirement will be assigned temporarily to those who have zero or lack entirely the @Sight property.

If multiple paths of the required length exist towards an object, then sight lines from all possible observer locations are shown.

If an object has a 3D model containing meta-information about the interesting parts of an object, then the geometrical center of this interesting area will be used as sight line target instead of the object’s insertion point.

Note that an object’s 2D symbol may be graphically displaced in relation to its insertion point, the ‘real’ XYZ-position. This can be seen as a dashed line from a selected object to its insertion point. The sight line calculations are unaffected by such ornamental 2D symbol displacements. The graphical displacement can be manipulated using the object’s square grip point.

## RC-DISPLAYTRAIN

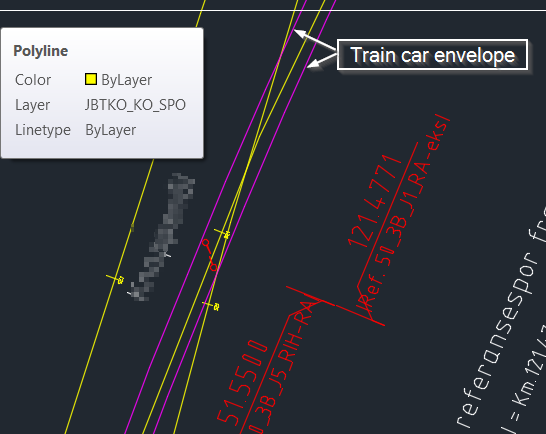
**Places a train of rectangle shaped cars oriented by RailCOMPLETE alignment objects.**

RC-DisplayTrain animates a train running through the connected alignment model as you move the CAD cursor. You may specify custom values for car width, bogie/axle separation, nose/tail extension. The running direction can be changed at any moment, in order to carry out shunting movements.

## RC-DISPLAYTRAINENVELOPE

**Calculates the outline or "envelope" of a train car passing through the infrastructure.**

The user is asked to select a track, and enter car dimensions.



The envelope is shown as purple lines in the drawing.

## RC-EDITPROPERTY

**Sets a property to a given value for one or more RailCOMPLETE objects.**

## RC-EDITSETTINGS

**Opens the settings window.**

## RC-EDITTABLE

**Edit an existing RailCOMPLETE table.**

## RC-EXPORT3D

**Export selected objects to 3D DWG file.**

## RC-EXPORTALIGNMENTSTOLANDXML

**Exports RailCOMPLETE document to LandXML format.**

## RC-EXPORTREFERENCEALIGNMENTDEFINITIONS

**Export reference alignment definitions from current AutoCAD document.**

## RC-EXPORTROUTES

**Export all possible elementary train routes into a new Excel document.**

## RC-EXPORTTABLETOEXCEL

**Export selected tables to Excel**

## RC-IMPORTALIGNMENTSFROMDWG

**Imports alignments from a DWG file to RailCOMPLETE document.**

## RC-IMPORTALIGNMENTSFROMLANDXML

**Imports alignments from a LandXML file to RailCOMPLETE document.**

## RC-IMPORTREFERENCEALIGNMENTDEFINITIONSFROMXML

**Imports reference alignment definitions from an XML file.**

## RC-JOINALIGNMENTS

**Joins two or more RC-ALIGNMENTS. Erases the old alignment, transfers all alignment data, and attached objects to the new alignment.**

## RC-LOADOBJECTTYPEDEFINITIONSFROMDRAWING

**Loads the Object Type Definitions (OTD) saved in the drawing and sets it as the current active OTD in computer memory. Consider backing up your drawing first.**

The current OTD in memory determines how new objects are inserted into the drawing. Please note: When switching active document (CAD system drawing) to a RailCOMPLETE document, the OTD is automatically re-loaded from the drawing.

## RC-LOADOBJECTTYPEDEFINITIONSFROMXML

**Loads new Object Type Definitions (OTD) from the selected XML file and sets it as the current active OTD in computer memory, without affecting the stored drawing or the objects’ current status in memory. Consider backing up your drawing first.**

The current OTD in memory determines how new objects are inserted into the drawing. Please note: When switching active document (CAD system drawing) to a RailCOMPLETE document, the OTD is automatically re-loaded from the drawing.

Read more about objects, tables, styles, 2D symbols and object type definitions in the sections above.

## RC-LOADRIBBON

**Starts RailCOMPLETE ribbon menu.**

Use this command if the RailCOMPLETE ribbon menu does not show after AutoCAD start-up.

## RC-LOADSTYLEDEFINITIONSFROMXML

**Loads StyleDefinitions from the selected XML file. Consider backing up your drawing first.**

Style definitions describe the details of alignment annotations, CAD system layer properties, fonts, linetypes etc. The administration’s default style definitions file came with the installation of RailCOMPLETE. Loading a different style definitions file will modify the appearance of several RailCOMPLETE tools as well of the objects created thereafter.

## RC-LOADSYMBOLLIBRARYFROMDWG

**Deletes all 2D symbols from the current drawing and loads new symbols from the selected DWG file. Consider backing up your drawing first.**

A 2D symbol library is a file with CAD system blocks. A default library comes with the installation of RailCOMPLETE, one for each represented railway administration.

Loading a different symbol library will modify the appearance of both existing and new objects. Loading a symbol library which is incompatible to the drawing’s current Object Type Definitions (OTD) gives unexpected results and may destroy your drawing irreversibly. Updating the symbol library also resets the list of drawings to ignore when "Ignore Missing Models" is turned on in 3D export.

2D symbols act as visual placeholders in your drawing, representing point objects. The symbols’ CAD system block names must correspond to the names used in the OTD for each variant within an object type declaration.

### 2D SYMBOLS USUALLY CONTAIN GRAPHICS ON SEVERAL CAD LAYERS

A symbol will be included in a unique CAD system anonymous block and assigned to a specific layer at its object’s creation, along with possible CAD system text attributes declared in the OTD for the relevant object type. Inside a symbol’s CAD system block there may be components which reside on other layers. Such CAD system layers can be turned on or off to display various characteristics of a symbol, e.g. “track appearance”, “signaling appearance” or “catenary appearance” when dealing with a symbol for a switch. The coloring of symbols is manipulated by several tools within RailCOMPLETE.

### 2D SYMBOL PARTS

The anonymous block which is instantiated for a given object type and variant at the object’s creation may contain several other drawing elements along with the basic 2D symbol. These drawing elements are also blocks, to be found in the symbol library file. E.g., “insertion point” showing where the local “0,0” block insertion point is, as well as several “magnetic point” blocks with suitable captions placed inside the object’s anonymous block. These “magnetic points” are handy for clicking other objects at certain specified positions, e.g. predefined positions for isolated joints, point machines, tongue rollers etc. in a switch symbol.

### CHANGING 2D SYMBOL WHEN VARIANT IS CHANGED

Changing an object’s variant will usually result in a change of 2D symbol, e.g. changing a 3-light main signal into a 4-light signal. The selection process and correspondence between object types, variants and their symbols are defined in the OTD.

### TEXT ATTRIBUTES IN 2D SYMBOLS

CAD system text attributes can be defined in the OTD for every object type and variant. Each such CAD text attribute is linked by attribute name to one of the object’s generic or custom properties. The current value of that property is displayed along with the 2D symbol, and is refreshed whenever there is a change in that property. The default position for each text attribute within the object’s anonymous block is defined in the OTD. There are several methods available for modifying the position, orientation and appearance of such text attributes after the object has been created.

## RC-MANAGEALIGNMENTS

**Toggles the Alignment Manager tool on/off.**

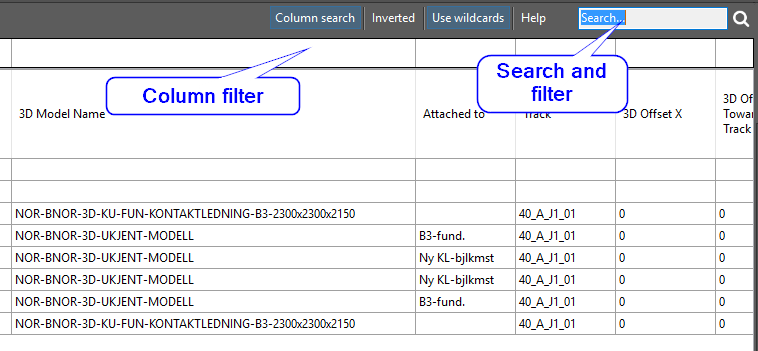
Alignments are all sorts of curved-line objects in 3D space, such as tracks, catenary wires, cables, cable ducts, roads, hand-rails etc. An alignment must have a geometrical description in the XY plane. It may have a profile description (vertical), defining the Z coordinate for point on the alignment (otherwise it will implicitly be zero). Positions along the alignment are measured from start to end in the XY plane. Additionally, alignments may have cant information, defining a rotation at every position. The Alignment Manager lets you view and to some extent also add, delete or modify information relative to alignments.

## RC-MANAGEOBJECTS

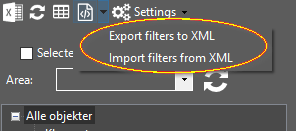
**Toggles the Object Manager tool on/off.**

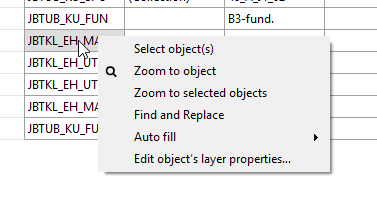
The Object Manager is organized somewhat like a spreadsheet, presenting objects row by row, and their properties column by column. You may edit, copy-paste-autofill, search and replace values, you may filter and sort object rows, and you may add, omit or reorganize property columns. The object manager’s contents can be exported to Excel, or inserted as a CAD system table into the drawing, for further refinement there. You may target objects in the Object Manager and zoom to them in the drawing, or you may select objects in the drawing and open the Object Manager with focus on that object.

The Object Manager is organized somewhat like a spreadsheet tool, allowing you to hide and unhide rows and columns through a variety of intuitive techniques. You can search, sort and filter. You can create, save and retrieve custom filters for later use or for sharing with other users.



You may easily switch between your CAD system’s model space and the Object Manager. The model space, the Object Manager and the RailCOMPLETE Properties tool are always synchronized. Objects may be selected in model space and acted upon in the Object manager, and vice versa.





A group of objects may be selected using the Object Manager or model space and then acted collectively on as a group using the Properties tool. You may set up filtering and sorting and then single-step through the resulting objects one by one, while you zoom to the current object in model space - useful for a quick assessment of your objects.

### OBJECT CATEGORY BROWSING

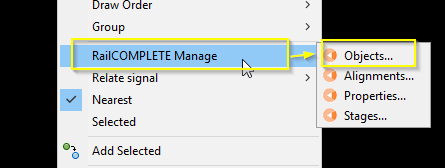
In the left part of the Object Manager window there is an object category browser, letting you turn on or off whole groups or subgroups of objects. This nomenclature has been defined in your railway administration’s Object Type Definitions (OTD) file.

### AREAS

In the upper left part of the Object Manager there is an Area reference field. When this field is non-empty, only objects being captured by that RailCOMPLETE area are subject to be shown or further filtered.

### SELECTED IN DRAWING

Alongside the Area field, there is a check-box for “Selected in drawing”.



By checking this box, the current filtered set of objects in the Object Manager is further narrowed down to only those objects that are also selected in model space.

### SELECTION AND MARKING

We distinguish between SELECTION and MARKING. A selected object means an object that is part of the CAD system’s current selection set, normally being highlighted with grips in model space. A marked object means an object with one or more gray-marked cells in the Object Manager window. Marked cells can be acted on in the Object Manager, but the CAD system has no notion of marked objects. Marking of objects is in use e.g. when picking out cells for Auto Fill or for narrowing down the current filter (buttons on the lower right).

### MODIFYING CELL VALUES

Modifying a cell value can be done either through direct data entry, or through the Search and Replace tool, or the Auto Fill tool. Cells that are permanently grayed down are non-editable. The reason may be that the objects containing these cells reside in a cross-referenced file (XREF), or that the object’s CAD system layer is locked against editing. Cell values may be modified even if the CAD system layer that the object resides on is frozen or turned off, I.e. even if you can’t see the object in the CAD system model space window. CAD system layer properties can be modified from the Object Manager.

### FORMULAS

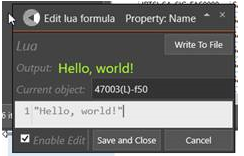
Cell values may be driven by formulas, much as in a spreadsheet. You can start editing a cell by moving the object manager focus to the desired cell and then hitting F2 or double-clicking in the cell. A formula editor symbol appears to the right in the cell.

RC-MANAGEOBJECTS

Clicking on this formula editor symbol starts the usual Lua formula editor tool.

### LUA EDITOR

The Lua editor lets you define your formula using straight Lua syntax and a multitude of built-in specialized RailCOMPLETE functions. When you are satisfied with your formula, you save and close the editor to return to the object manager. If needed, you can write your formula – which may actually be a very large and complex Lua program - to a file for later use, or for sharing with other users.



As an alternative to using the Lua editor tool, you can enter your formula directly in the object manager cell by starting with an equality sign ‘=’ followed by your formula.

RC-MANAGEOBJECTS

When saving and closing the Lua editor, or when pressing ENTER after in-cell editing, you will see that a formula symbol appears to the left in the cell.

RC-MANAGEOBJECTS

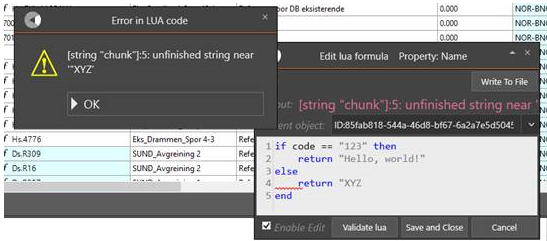
This (the “f”) shows that there is a formula behind the displayed cell value. Trying to type in a new cell value directly will no longer work, in order to protect your formula from being overwritten by accident. To edit the formula again, simply open the Lua editor or edit the formula directly in the cell. The other tools with access to properties (notably Manage Properties and Edit Properties) will also be blocked from accidentally overwriting a formula.

### REMOVING FORMULAS

In order to remove a formula, you re-enter cell edit mode and replace the Lua formula with an entirely blank formula. Alternatively, you may right-click and use the “Replace formula with cell value” function. The previously shown value – the result of the formula – will now replace the formula. If the formula previously had an error or evaluated to a blank cell, then the cell value will now be empty.

### DRAWING ATTENTION TO A VALUE – WARNINGS AND CEHCK MARKS

If there is an error in a formula, then the Lua editor will warn you and also help you find the location and the cause for the error. In many cases, you will see that warning triangle appears in a cell. The reason may vary, but this indicates a problem, such as missing or bad input values to the formula. In order to make a formula return a value x and at the same time show the warning triangle, use this Lua statement: “return x, \_warning” instead of just “return x”. You may return any of the following attention symbols: “\_warning” (a warning triangle), or “\_ok” (a check mark).



Formulas with Lua syntax errors show up in the object manager cell as blank cells, but with a warning sign along with the formula symbol.

RC-MANAGEOBJECTS

### EXPORT OBJECT DATA TO EXCEL

The Object Manager allows for easy export to Excel. Set up your column selection by clicking in the top row and adding (unhiding) or deleting (hiding) columns. Drag columns sideways to organize your view as needed. Add area, filters and search patterns to narrow down the row selection. Then apply sorting by clicking on the relevant columns in the reverse sorting sequence. Click on the Excel export button, then the current view is exported to a new spreadsheet including column titles and respecting the sorting order.

### CREATING TABLES FROM THE OBJECT MANAGER

The Object Manager is also ideal as a flying start for producing high-quality tables. Set up columns and rows as for the Excel export, then click on the RailCOMPLETE Table button. The RailCOMPLETE Table Manager tool opens, letting you refine the table definition further, save the table definition, and finally create the table in model space. Export from this refined table definition is then available from the Table Manager.

## RC-MANAGEPROPERTIES

**Toggles the Properties Manager tool on/off.**

The Properties Manager displays property names and values, line by line. Objects which are currently selected in the CAD system will be presented with “VARIES” instead of a specific value if two or more object’s values differ for a certain property. A property’s value may be the result of an underlying formula, which is indicated with a formula symbol to the left of the evaluated value (or VARIES). If more than one object has been selected and there are different formulas for a certain property, then an asterisk “\*” will be shown instead of the formula symbol, also when the formulas all evaluate to the same value. See also the Edit Property command.

## RC-MANAGESTAGESUSINGLAYERNAMES

**Provides an interface for turning the CAD system layers on and off in order to display stage dependent drawing details for objects that reside on the affected layers.**

Objects may be RailCOMPLETE objects or any type of CAD object (circle, line, block, leader, text, XREF etc). Specify the stage pattern as either a - stage pattern, or as a stage pattern, with user defined prefix, entering/leaving infix and suffix patterns, number of digits used for denoting main and sub stage numbers, as well as a user defined main/sub stage separator. For instance, the pattern “MyPrefix”, “2”, “.”, “2”, “-”, “MySuffix” refers to all layers whose names contain “MyPrefix##.##-##.##MySuffix”, where # denotes any single digit 0-9.

## RC-MATCHPOSITION

**Matches the position of the selected objects to a target object, setting their mileage to the same value.**

You may direct the command to use the other object’s mileage as derived from its own alignment, or the other object’s reference mileage. You may choose to match the other object’s alignment name and XY-position without altering Z, or you may match “everything”, i.e. alignment name and XYZ coordinates.

## RC-MOVESELECTIONALONGPATH

**Move selection of objects along a path, based on input parameters.**

Items can be moved along (reference) alignment.

## RC-MOVETEXTATTRIBUTES

**Moves the text attributes for the selected objects as a group.**

## RC-OPENXREF

**Opens the Xrefed RailCOMPLETE document in a new window.**

## RC-P000001-UPDATELABELS

**Patch which removes .labelItemDefinition from label objects.**

## RC-P000002-UPDATETABLES

**Patch which replaces .attachment/.area with attachment/area reference.**

## RC-P000003-MOUNTING3D

**Patch which copies Mounting3D value into PoleRouting3D.**

## RC-P000004-FIXMILEAGE

**Patch which moves mileage data (mileage at start and mileage changes) from railML to LandXML element.**

## RC-P000005-FIXCONNECTIONS

**Patch which moves connections to new container, which allows for multiple end connections.**

## RC-P000006-CHANGEDATATYPEOF68SIGNALS

**Patch which changes railML datatype of NO-BN “Signal 68” objects from tSpeedChange to tSign.**

## RC-P000007-RENAMETRACKTOALIGNMENT

**Migrate data after renaming track=>alignment throughout the code.**

## RC-P000010-MOVEOFFSETFORMULAS

## RC-PEEKATOBJECTDEFINITION

**Peek at the XML code behind a RailCOMPLETE object.**

Although you may modify an object’s stored definition using this tool, it is strongly advised not to do so unless you are an expert user. Please consult the RailCOMPLETE Agent’s User Manual.

## RC-PEEKATOBJECTTYPEDEFINTIONS

**Peek at the XML code defining how new objects are created, interrelated and represented symbolically.**

The Object Type Definitions XML file contains an OTD identification header and information about fonts, linetypes, default file system folders, as well as detailed descriptions of object types needed for a given railway administration, with symbols, properties, methods, relations and create-time default values.

Although you may modify a drawing’s “genetic information” using this tool, it is strongly advised not to do so unless you are an expert user.

Please consult the RailCOMPLETE® Agent’s User Manual.

Several built-in generic object types are available with a multitude of properties and methods. Among the common basic properties, you will find the object’s id, name, description, position along a specified alignment, its relative elevation and its distance to this alignment. An alignment may be a track’s center line, a road center line, a cable duct, a catenary wire etc. The built-in methods can manipulate the object further in 3D space, associate it with a 3D model, draw poles between boards and foundations etc. There are also methods for obtaining all sorts of information from the alignment that a point object has been connected to, such as elevation, gradient, curvature and cant.

You may peek at zero-dimensional point objects, one-dimensional alignment objects and two-dimensional area objects.

Specialized objects are declared in the OTD, starting out with one of the generic object types available. The object type declaration adds as many custom properties and methods as needed. Each object type declares its own Attachment Category and its affinities for other such categories, allowing such an object instance to attach to another suitable object and thereby become part of a daisy-chain of physical object positions in 3D space.

Each object type also declares its own Relation Source Space, allowing it to be logically related to other objects. The OTD can hold an arbitrary number of binary relations.

Every object type can be declared with several Variants, each with its own recognizable 2D CAD symbol. For each variant there may be several custom properties influencing the object’s description, such as width, length, height, number, color etc. Each relevant combination of custom property values can be associated with a 3D model for this object.

When an object is created, there may be standard values declared in the OTD which are assigned to object properties. E.g., this may be a board’s standard elevation above its foundation, or an axle counter’s standard placement at +/-0,75m from track center line, etc.

In many cases there are no definite values to standardize on in the OTD, but there may still be standard methods. E.g., a balise will always be mounted in the center track, but it must be lifted to half of the local track superelevation, as well as pitched with the local track gradient and rolled with the local track cant. In switches you may even find balises that are yawed in relation to their track’s center line, because the sleepers may follow the switch’s other leg’s orientation. In such a case, the OTD will declare a Formula to be evaluated for the relevant property, just as you would do in a spreadsheet. The superelevation formula would read something like “getAlignmentInfo().Superelevation/2000”, assuming that superelevation is expressed in millimeters, and be applied to the object’s Z-coordinate property.

## RC-PEEKATTABLEDEFINITION

**Peek at the definition behind a RailCOMPLETE table.**

Although you may modify a table’s stored definition using this tool, it is strongly advised not to do so unless you are an expert user. Please consult the RailCOMPLETE Agent’s User Manual.

## RC-QUANTIZEPOSITION

**Quantizes the selected object’s absolute position.**

Specify either the object’s alignment or its reference alignment mileage to be quantized. The quantization value may be any number Q, mapping a value V into floor((V + Q/2)/Q)\*Q. For instance, V=16.221 and Q=2.5 will map to 15.0, whereas V=16.267 and Q=2.5 maps to 17.5.

## RC-QUERYOBJECT

**Opens a Query Object window where you can access all drawing information that is linked directly or indirectly to the selected object.**

The Query tool uses the built-in Lua programming language in order to access information from the drawing with the currently selected object as a starting point. For instance, let’s assume that the object that was selected is a turnout with “512” as its @code property value. Typing code will show 512. Typing ”V.”..code will show V.512, etc. Note that Lua is generally case-sensitive.

## RC-QUICKSELECT

**Quick Select**

Add objects to the current selection set. Select objects based on the properties @Name, @Id, @Code, @Type, @Position. With Position, you will be prompted for XY coordinates as well as a radius. With V1.0, coordinates must be entered as World coordinates.

## RC-REFRESHOBJECTS

**Updates attachments and refreshes property values derived from attachments.**

Note that refreshing an object may affect other objects because they might depend on each other. The refresh depth can be adjusted.

## RC-REFRESHTABLE

**Refreshes a RailCOMPLETE table.**

The current table settings, including object filter, the Lua formula for each column, column widths, column title, title wrapping etc are evaluated over again. Please note that the objects which constitute the rows in a table are not themselves refreshed, so be sure to refresh all the concerned objects before you refresh a table.

## RC-RELATIONUPDATEDEPTH

**Set the depth in of which objects will be updated when an object is changed. The depth of an object is the number of relations to get to the changed object.**

## RC-RELOADALLXREFS

**Reloads all Xrefs.**

## RC-RELOADOBJECTMANAGERDATA

**Reloads all objects in Object Manager.**

## RC-RELOADXREF

**Reloads selected Xrefs.**

## RC-REMOVECONNECTION

**Prompts the user for a switch or alignment. If the object contains multiple connections, the closest connection will be removed. Does not work for objects in XREFs.**

## RC-REMOVEPATHS

**Removes all paths in drawing.**

## RC-REMOVEUNKNOWNDATA

**Removes any railML "any" elements/attributes from document.**

## RC-RESETLAYER

**Updates layers to include any changes in the layer specifications from the current document's OTD**

## RC-RESETLUAFORMULAS

**Updates the Lua formulas in the selected objects according to their object type definition.**

## RC-RESETTEXTATTRIBUTES

**Resets the object’s text attributes’ positions, text orientations and obliquing to their original create-time values.**

Only position, orientation (rotation) and oblique angle are reset. There are many other aspects of attribute text appearances that will not be affected, for instance:

• Justification - what part of the text that position refers to (top left (TL), middle center (MC) etc)

• Text scale

• Width factor

• Upside-down

• etc

To adjust these aspects for only one of your object’s text attributes at a time, you should bring up AutoCAD’s Enhanced Attribute Editor (EATTEDIT) by double-clicking on one of your object’s texts, select the individual text that you need to adjust, and switch to the Text Options tab.

In order to simultaneously adjust all text attributes for the selected objects(s), use the AutoCAD \_SCALETEXT and \_JUSTIFYTEXT commands. Note: The starting underscore forces AutoCAD to use the native version of the command, in case your organization has defined a company-specific command overrule.

## RC-REVERSEALIGNMENT

**Reverses the selected alignments.**

## RC-RUNSCRIPT

**Executes direct commands in order to create new drawings or modify existing ones, or export data in various formats.**

## RC-SAVEOBJECTTYPEDEFINITIONSINDRAWING

**Saves the current Object Type Definitions (OTD) held in memory to the drawing’s header data structure and to the drawing’s individual objects, but without mapping deprecated property values to updated values. Consider backing up your drawing first.**

The current OTD in memory is made permanent in the drawing and in its objects. The OTD will reappear in computer memory as an active OTD each time the drawing is closed and re-opened, and each time the user switches active document and switches back to the RailCOMPLETE drawing.

Please note that no property values are updated to new enumeration sequence values with this save OTD method.

If existing property enumeration values shall be changed to new enumeration values, and if new object properties and CAD text attributes are to appear in already created objects, then the “update OTD” command must be used instead.

## RC-SELECTBYHANDLE

**Sets selection by given AutoCAD handle. Optionally zooms to object.**

## RC-SELECTSIMILAR

**Select all objects with the same types as already selected objects.**

## RC-SETALIGNMENTGEOMETRYFROMPOLYLINE

**Replaces the alignment’s existing horizontal geometry with a 2D polyline as horizontal geometry.**

Please note that AutoCAD 2D polylines consist of piecewise circular arcs and straight lines. Clothoids may therefore not be updated in using this command, but will be replaced by a sequence of circular arcs.

## RC-SETBLOCKCOLOR

**Lets the user set the color of a selection of blocks**

## RC-SETLAYERPROPERTIES

**Manage AutoCAD layers for selected objects**

## RC-SETLENGTHPROPERTY

**Adds an alignment property (service section, tunnel, bridge, platform, or unprotected area) to the currently selected path (see**[**RC-SETPATH**](file:///C:\Users\andar\AppData\Roaming\Autodesk\ApplicationPlugins\RC.bundle\UserGuide\commands\index.html#rc-setpath)**).**

## RC-SETOBJECTTYPEDEFINITIONS

**First time: Upgrades a plain DWG file to a RailCOMPLETE document. Subsequent use: Provides maintenance for the document’s object type definitions.**

Adds a RailCOMPLETE document header to the current DWG file.

Select the desired railway administration and Object Type Definitions among the available choices. The object type definition choice determines the kind of objects that can be inserted in a document, their properties, symbols, 3D models, relations etc.

Once the RailCOMPLETE document header has been added, the START-icon changes to the chosen administration’s logo.

## RC-SETPATH

**Sets the current path on railway alignments in RailCOMPLETE document.**

## RC-SETPATHSFROMAREA

**Set RailCOMPLETE path to intersection of area and selected alignments.**

## RC-SETTEXTATTRIBUTEORIENTATION

**Changes the text attribute orientation in the 2D-symbol for the selected object(s), orienting object text attributes along the object’s local alignment, the current user coordinate system, the world coordinate system or the current screen view.**

## RC-SHOWALIGNMENTNAME

**Displays transient railway alignment name markers along the selected railway alignments.**

## RC-SHOWATTACHMENTLINES

**Displays transient connection lines between relations and attachments.**

## RC-SHOWDISTANCETOALIGNMENT

**Shows the distance from a selected point object to its alignment.**

If the «Ref» alignment button is active, the distance to the reference alignment is shown. If no point object is selected, the current CAD cursor’s position is shown.

## RC-SHOWELEVATIONABOVEALIGNMENT

**Displays the CAD cursor’s or a point object’s elevation above sea level.**

If a point object is selected, then its elevation is decomposed into the alignment’s elevation, the object’s relative elevation above the alignment, and the absolute elevation (which coincides with the CAD system Z coordinate). If no point object is selected, the current CAD system default Z coordinate is used as a basis. If the «Ref» alignment button is active, the elevation above the reference alignment is shown.

## RC-SHOWFOULINGPOINTS

**Displays fouling points.**

## RC-SHOWGEOMETRY

**Displays transient alignment geometry markers along the selected alignments.**

## RC-SHOWHELP

**Opens the Help window.**

The help window can also be opened by pressing F1 when the mouse pointer is hovered over an icon.

## RC-SHOWLAYERS

**Shows or hides predefined layer groups. Layer groups are configured in the relevant administration’s OTD (not the AutoCAD layer manager groups). To delete, add or modify layer such groups, contact your local administrator or the relevant RailCOMPLETE agent.**

## RC-SHOWMILEAGE

**Displays transient mileage markers along the selected railway alignments.**

## RC-SHOWMILEAGECHANGE

**Displays transient mileage change (aka «chain break») symbols along the selected alignments.**

## RC-SHOWMODULES

**Shows the Modules form. When closed, load new modules and set the startup modules setting.**

## RC-SHOWPOSITION

**Displays the transient mileage indicator line along with a mileage cursor as you move the CAD cursor around.**

The mileage indicator line is a line being perpendicular to the alignment that the mileage cursor is currently following.

The little square symbol shows which alignment is currently used as data source.

The mileage shown is the same as the “own alignment mileage” for any object located on the mileage indicator line.

The mileage cursor is split in a little circle following the “own alignment” and the square one for the reference alignment whenever [RC-ShowPositionReferenceAlignment](file:///C:\Users\andar\AppData\Roaming\Autodesk\ApplicationPlugins\RC.bundle\UserGuide\commands\index.html#rc-showpositionreferencealignment) has been activated (the “Ref” button).

By default, the position will use the alignment being nearest to the CAD cursor as data source.

By activating [RC-SnapToSelectedAlignment](file:///C:\Users\andar\AppData\Roaming\Autodesk\ApplicationPlugins\RC.bundle\UserGuide\commands\index.html#rc-snaptoselectedalignment) (the “Snap” button), the mileage will use the last selected alignment (even if deselected) as data source.

## RC-SHOWPOSITIONINALIGNMENT

**Displays transient connection lines between relations and attachments.**

## RC-SHOWPOSITIONREFERENCEINALIGNMENT

**Activates the “Ref” ribbon button. The mileage cursor is split in a little circle following the “own alignment” and a square one for the reference alignment. Reference mileage is indicated whenever the Show Position (the mileage indicator line) has been activated.**

## RC-SHOWPROFILE

**Displays transient profile markers along the selected railway alignments.**

## RC-SHOWREFERENCEALIGNMENT

**Displays transient reference railway alignments along the selected railway alignments.**

## RC-SHOWREFERENCEMILEAGE

**Displays transient mileage markers along the selected railway alignments, but only in the intervals where the railway alignment is its own reference.**

## RC-SHOWRELATIONLINES

**Displays transient connection lines between relations and attachments.**

## RC-SHOWTOPOLOGY

**Displays topology data.**

An open-ended alignment (unconnected) is shown with an angle bracket ‘<’ or ‘>’. An alignment end which is connected to another alignment is shown with a circle.

To remove an existing alignment connection, use the [RC-REMOVECONNECTION](file:///C:\Users\andar\AppData\Roaming\Autodesk\ApplicationPlugins\RC.bundle\UserGuide\commands\index.html#rc-removeconnection) command.

## RC-SHOWTUTORIALS

**Toggles between a single-line and a two-line representation of all alignments of type ‘track’.**

## RC-SHOWTVDSECTIONSTRANSIENT

**Displays transient lines between delimiters in a tvd section.**

## RC-SHOWTWORAILS

**Toggles between a single-line and a two-line representation of all alignments.**

## RC-SHOWVERSION

**Prints the current RailCOMPLETE installation's version to the CAD command window.**

## RC-SHOWWARNINGINCADTEXTWINDOW

**0: Write to command line 1: Show dialogs**

## RC-SNAPTOSELECTEDALIGNMENT

**Activates the “Snap” ribbon button. The mileage indicator line will use the last selected alignment (even if deselected) as data source.**

## RC-SWITCHBLOCK

**AutoCAD command to replace a BlockReference by another BlockReference with different BlockDefinitions, preserving as much text and position information as possible.**

## RC-TESTER-AUDITSWITCHCONFIGURATION

**Ensure that the configuration of the selected switch matches the track geometry around its insertion point.**

## RC-TESTER-DRAWINFRASTRUCTUREVISUALIZATIONS

**Draw the schematic representation of the current infrastructure, represented by "clone objects". The schematic objects are drawn in the current drawing with respect to the UCS.**

## RC-TESTER-EXPORTTORAILML

**Exports objects to railML format.**

The railML export feature is under development and has not yet been certified by railML.org.

## RC-UNLOADXREF

**Unloads selected Xrefs.**

## RC-UPDATEAREA

**Reloads the Areas in the drawing.**

## RC-UPDATEOBJECTTYPEDEFINITIONSWITHMAPPING

**Updates a RailCOMPLETE document to be compatible with a new Object Type Definitions (OTD) version, according to the selected OTD and mapping files.**

The update process is based on an external OTD together with a mapping file describing how deprecated properties shall be mapped onto new or other existing properties. The mapping file also describes how predefined but deprecated values shall be mapped into new or existing other values.

The old and the new OTD must be intended for the same railway administration. A new OTD file can made with manual or tool-based editing of the old OTD file.

The mapping file must be created using the dedicated RailCOMPLETE command.

## RC-UPDATESINGLELANDXMLALIGNMENT

**Ask user for LandXML file and a specific alignment in the LandXML file. Copy this alignment to the selected railway alignment.**

## RC-WEBPAGESANDFAQ

**Opens an information resources page with useful info and links.**

## RC-XREFSELECTOFF

**Turns off selection mode for selecting components within a Xref. The Xref will be treated like one separate object.**

## RC-XREFSELECTON

**Turns on selection mode for selecting components within a Xref. In order to select the Xref itself, use the ‘green/blue’ or the lasso selection method in AutoCAD to swipe an area somewhere over the Xref.**