TopSky plugin for EuroScope

- version 2.2 -

Developer Guide

EuroScope | power of control

Table of Contents

1 EuroScope setup 3

1.1 General Settings 3

1.1.1 Display options 3

1.1.2 TAG display options 3

1.1.3 Miscellaneous options 3

1.2 Display settings dialog 3

1.3 Symbology dialog 4

1.3.1 Colors 4

1.3.2 Symbols 4

1.4 Plug-ins Dialog 4

1.5 Conflict Alert Settings Dialog 4

1.5.1 STCA Lower altitude 4

1.5.2 STCA Higher level 4

1.5.3 MTCA options 4

1.6 Plugin data files and internal settings 4

1.7 TAG editor 5

2 External data files 6

2.1 TopSkyAirspace.txt 7

2.2 TopSkyAreas.txt 9

2.3 TopSkyAreasManualAct.txt 14

2.4 TopSkyCPDLC.txt 15

2.5 TopSkyCallsigns.txt & TopSkyCallsignsLocal.txt 18

2.6 TopSkyMaps.txt & TopSkyMapsLocal.txt 19

2.7 TopSkyMSAW.txt 26

2.8 TopSkyRadars.txt 28

2.9 TopSkySettings.txt & TopSkySettingsLocal.txt 30

2.10 TopSkySSRcodes.txt 31

2.11 TopSkySTCA.txt 39

2.12 TopSkySymbols.txt 42

2.13 TopSkyViews.txt 44

1 EuroScope setup

This chapter is meant as a guide for users who either didn't get the plugin as a part of a package including all the settings files, or who want to create their own setup.

To load the plugin and setup EuroScope to make the most out of it, use the following settings in the EuroScope settings dialogs. Settings not mentioned shouldn't make any difference. Remember to save the settings when exiting ES. Use of the plugin file must be saved to each profile file separately for it to be loaded automatically.

1.1 General Settings

1.1.1 Display options

- Show route when accepting Off
- Lock show route when accepting Off
- Rotate radar target symbol Off
- Rotate flight plan track symbol Off
- Show CLAM warnings On
- Show RAM warnings On

The plugin uses its own route display and the default plugin track labels don't offer any way of hiding the ES route display so the first two should be selected off. If both settings are left "on" and the track labels are not modified, the radar screen will soon be filled with aircraft routes with no way to remove them. The CLAM and RAM warning settings also affect the plugin. If they are selected off, the plugin can't show the warnings either.

1.1.2 TAG display options

- Allow correlated aircraft tag untagged On
- Allow concerned aircraft tag untagged Off
- Allow assumed aircraft tag untagged Off
- Show detailed over untagged On

These settings are needed to display the correct type of track label for each aircraft.

1.1.3 Miscellaneous options

- Keep scratch pad content after direct On

The plugin stores the manual alerts in the scratch pad. If this setting is selected off and a direct clearance is given, any manual alerts are removed as a result.

1.2 Display settings dialog

- Number of history dots 0
- Show leader lines Off

The plugin draws its own custom history dots and leader lines so the ES default ones must be selected off.

1.3 Symbology dialog

1.3.1 Colors

Whether to use the transparency settings is up to the user, and their colors should be set as desired. The plugin colors determine the track label color but the EuroScope defined color is used to draw the leader line between the position symbol and the label. The “Colors” chapter in the localization part of the manual shows some colors having a corresponding ES color. In those cases the plugin and ES colors should be set to the same value.

“Other/normal menu item” and “Other/disabled menu item” should be different colors to be able to see disabled menu items in the setup menus.

“Sector/active sector background” and “Sector/inactive sector background” should be chosen so that all plugin colors can be seen. Medium grey colors work well with the default plugin colors but anything can be used as long as the plugin colors are taken into account and changed accordingly.

1.3.2 Symbols

The “Aircraft primary radar only”, “Aircraft flight plan track”, “Aircraft coasting”, “History dot” and all entries beginning with “Aircraft corr.” or “Aircraft uncorr.” are drawn by the plugin and should be set to “MOVETO 0 0” only. All other symbols are drawn by ES and should be set as desired.

1.4 Plug-ins Dialog

Load the plugin file (TopSky.dll). After loading it, highlight its name (“TopSky plugin”) in the list and move “Standard ES radar screen” from the “Forbidden to draw on types” box to the “Allowed to draw on types” box.

1.5 Conflict Alert Settings Dialog

The plugin has its own STCA and MTCD systems and doesn’t use the ES default ones. Selecting the below settings will disable the warnings from the ES default systems.

1.5.1 STCA Lower altitude

- Show lower altitude STCA Off

1.5.2 STCA Higher level

- Show higher level STCA Off

1.5.3 MTCA options

- Warn if vertical separation is less 0 feet
- Warn if horizontal separation is less 0 nm

1.6 Plugin data files and internal settings

The plugin comes with some pre-defined setups, but it is possible to adjust hundreds of different settings using the settings files. This makes it possible to update the existing setups and to create completely new ones. Even though the ranges of acceptable values have been limited to prevent major problems, care should be taken when adjusting any settings as the results can be unexpected.

When creating a new setup, only the TopSkyCallsigns.txt data file can be used from an old setup, as it contains only general callsign data that can be used anywhere. All the other data files contain information specific to the original setup, and should not be used.

Some setup-specific plugin settings are listed below. At least these settings should be looked at when creating a new setup, but many of the others are important too. However, never change a setting without understanding what it does.

Setup_COOPANS	Define which of the two plugin types to activate (A or B)
Airspace_C_Flag Airspace_C_No_Flag	Define the list of clearance flag airports
Airspace_P_Flag Airspace_P_No_Flag	Define the list of P-RNAV airports (plugin type B only)
Airspace_UsePluginASSR Airspace_ASSR_StartCode Airspace_ASSR_EndCode	Define the transponder code ranges
Airspace_VFR_Code	Define the VFR transponder code
Airspace_SIGMET_Areas Airspace_NOTAM_Add Airspace_NOTAM_Remove	Define airspace information for the SIGMET/NOTAM functions

1.7 TAG editor

To make the plugin work as it's supposed to, there are some rules that need to be followed when creating the tag families:

- When a tag item has an "**(unselected track)**" variant, it should be used in the *untagged* and *tagged* tagging levels.
- Every correlated and flight plan track *untagged* and *tagged* tag must contain the "**Dummy item – not detailed tag**" item. Their *detailed* tags must contain the "**Dummy item – detailed tag**" item.
- Items beginning with "**List**" should be used in flight lists, other items in tags ("List" items are not hidden when a label is subject to filtering, the other ones are)
- Items beginning with "**ET**" should be used in the *detailed* tagging level (they are only displayed in the extended label which is a sublevel of the *detailed* tagging level in the plugin, displayed by the "**Open extended tag**" function)
- Items with "**(0-1)**" are only displayed when the track is unconcerned or notified, items with "**(2+)**" when the track is in any other state
- A couple of the indicator items have an "**(inactive)**" version as well. Both can be used in the same tag, as only one will be shown
- Items with "**(not ET)**" will not be shown in the extended label

2 External data files

This chapter gives guidance on developing the external data files used by the plugin for various features. Even though the plugin does its best to check the data for errors, some errors may get through and cause all kinds of issues, possibly leading to ES crashing, so it's important to be careful to provide correctly formed data when creating the files. Errors found in the data files by the plugin can be seen in the "Plugin Status" submenu. It is found in the "Status" or "STS" menu depending on the plugin version.

To reload a data file, left-click on the "Reload" button. Some of the data can also be visually checked for correctness by left-clicking the "View" button. The areas will then be shown on the radar screen.

2.1 TopSkyAirspace.txt

The file contains the QNH value to transition level tables for the QNH/TL Window and any custom runway approach lines (i.e. starting point not on the runway threshold and/or approach course not aligned with the runway centerline) and lists of custom level menu values.

For the QNH/TL tables, the file is read one line at a time and the first matching line will be used. The following example shows a possible setup:

// Made up lines…	Comment
QNHTL:XBZZ:80	QNHTL
QNHTL:XB,XC:80,978,75,996,70	QNHTL

For the runway approach lines, any lines defined here will override the corresponding lines automatically created from the sector file data. The following line shows an example:

APPLINE:EFHK:15:153.4:HEL	AppLine

For the custom level lists, the file is read one line at a time and the first matching line will be used. The following example shows a possible setup:

LEVELS:EETN:22,25,32,35,50	LEVELS
LEVELS:*:15,20,30	LEVELS

QNHTL

QNHTL:ICAOlist:TL
QNHTL:ICAOlist:TL_1,QNH_1,TL_2,QNH_2, … ,TL_n,QNH_n,TL_{n+1}

Defines a variable transition level based on the QNH value for the specified airports.

- ICAOlist — List of airport ICAO designators, separated by commas (“,”). Either complete designators or one to three first letters of the designator.
- TL_n — Transition level value (text string, will be displayed exactly as written)
- QNH_n — QNH value (integer value, see also below)

The first definition sets a fixed transition level, while the second one creates a table of transition levels based on QNH values.

The QNH value must be in the same format as in the METAR. If the METAR reports the QNH in hPa (Qxxxx), the values in the list must be in hPa as well. If the METAR reports the QNH in inHg (Axxxx), the values in the list must be in inHg*100.

The TL/QNH list must contain one or more pairs of TL and QNH values followed by a TL value in the end.

The transition level is found in the following way: the actual QNH value found in the METAR is compared against the values in the list, from left to right. If the actual QNH is less than the list value, the corresponding TL value (the one before the QNH value) is used. If not, the next QNH value in the list is

checked and so on. If the actual QNH value is equal to or greater than all the values in the list, the TL value after the last QNH value in the list (TL_{n+1}) is used.

AppLine

APPLINE:AirportCode:Runway:AppCourseT:PointName
APPLINE:AirportCode:Runway:AppCourseT:Lat:Lon

Defines a runway approach line with a specified approach course and end point (either as a point name or coordinates). By default approach lines are created for all runways in the active sector file with the approach courses set on the extended runway center lines and ending at the runway thresholds. If a runway needs some other approach line instead, it can be defined here. Any runway approach line defined here will override the plugin created default one for the same runway.

- AirportCode — The airport ICAO code
- Runway — The runway identifier
- AppCourseT — The approach track (degrees true, decimal value)
- PointName — Fix, VOR, NDB, airport code or runway (must be found in the active sector file)
- Lat — Endpoint Latitude (decimal degrees or sector file format)
- Lon — Endpoint Longitude (decimal degrees or sector file format)

Note: the syntax to define a runway threshold as a PointName is the 4-letter ICAO airport designator followed by a forward slash and the runway identifier.

Levels

LEVELS:AirportCode:LevelList

Defines a custom level list to be used in all plugin level menus (AFL, CFL and RFL). The list here overrides the default list values up to the highest value in the custom list, whichafter the default list takes over. The airport code is compared against the login callsign, and if a matching list is found, it will be used. In case a matching list is not found, the code then compares the lists against the aircraft's departure airport until more than 1/3 of the way to the destination, and then against the destination airport.

- AirportCode — The airport ICAO code. "*" matches any airport, other wildcards are not supported
- LevelList — Comma-separated list of levels (in 100's of feet, range 1-999)

2.2 TopSkyAreas.txt

This file contains the areas for the APW and SAP functionality as well as the MTCD and STCA inhibit areas. The following example area is used to show the syntax (optional lines in grey color).

CATEGORYDEF:D:7:0:5:0:0:0	CategoryDef
// EF D101 Isosaari	Comment
AREA:T:EFD101	Name
CATEGORY:D	Category
ACTIVE:NOTAM:EFIN:EF D101	Active
ACTIVE:NOTAM:EFIN:EFD101	Active
LABEL:N059.55.08.817:E025.07.08.496:D101	Label
LIMITS:0:390	Limits
N059.56.20.000 E025.16.53.000	Coordinate
N059.54.15.000 E025.15.06.000	Coordinate
N059.53.27.000 E024.59.49.000	Coordinate
N059.56.36.000 E025.10.10.000	Coordinate

CategoryDef

CATEGORYDEF:Name:ActBorderColor:ActFillColor:ActFillPattern:PreBorderColor:PreFillColor:PreFillPattern
CATEGORYDEF:Name:ActBorderColor:ActFillColor:ActFillPattern:PreBorderColor:PreFillColor:PreFillPattern:LabelName:LabelMapText:LabelUserText:LabelLevels:LabelTimes

This line defines the display colors for an area category. The default border color is *Active Map* for unfilled and *Active RD Map* for filled areas. The default fill color is *Active RD Infill Map*, and the fill percentage that was used in previous plugin versions is 50. If the label settings are left out, the default settings are used.

- Name — Name for the category (text string)
- ActBorderColor — Active area border color
  - 0 — use default color
  - 1-20 — custom color (*Active Map Type X*)
- ActFillColor — Active area fill color (values as in ActBorderColor)
  - 0 — use default color
  - 1-20 — custom color (*Active Map Type X*)
- ActFillPattern — Active area fill pattern
  - 0 — no fill
  - 5, 10, 20, 25, 30, 40, 50, 60, 70, 75, 80, 90, 100 — percentage to fill
- PreBorderColor — Pre-active area border color (values as in ActBorderColor)
- PreFillColor — Pre-active area fill color (values as in ActFillColor)
- PreFillPattern — Pre-active area fill pattern (values as in ActFillPattern)
- LabelName — Show area name in area label (1=yes, 0=no)
- LabelMapText — Show map text in area label (1=yes, 0=no)
- LabelUserText — Show user text in area label (1=yes, 0=no)
- LabelLevels — Show area level limits in area label (1=yes, 0=no)
- LabelTimes — Show area activation times in area label (1=yes, 0=no)

Name

AREA:AreaType:AreaName

The first line for each area definition must be a name line. Type S areas will appear in the Safety Nets Status Window, type M areas in the MTCD Status Window and the other types are in the TSA Areas window.

- AreaType — Area type (one or more of the following):
  - T (TSA area)
  - 1, 2, 3, 4 or 5 (TSA area)
  - 1F, 2F, 3F, 4F or 5F (TSA area)
  - M (MTCD inhibit area)
  - S (STCA inhibit area)
- AreaName — Area name to identify it in the areas window (text string)

TSA area types 1F-5F are filled, 1-5 are not (the number defines the area border color). For area type T the colors and filling are defined in a Category line. If a category line is not defined for a type T area, the area is not filled and a default border color is used. For compatibility with earlier plugin versions, area types D and R are also supported (they are TSA areas that are filled)

Area type T and the category definitions are meant to be used with plugin type A and area types 1-5 and 1F-5F with plugin type B, but their use is not restricted by the code.

Category

CATEGORY:Name

The category line is optional. It defines the display colors for type T areas.

- Name — Category name (must have been defined earlier in the file)

Label

LABEL:Lat:Lon:LabelText

The label line is optional. It displays the area name and/or altitude limits on the radar display.

- Lat — Latitude for the label (decimal degrees or sector file format)
- Lon — Longitude for the label (decimal degrees or sector file format)
- LabelText — The label text (text string)

Active

ACTIVE:1
ACTIVE:SchedStartDate:SchedEndDate:SchedWeekdays:StartTime:EndTime
ACTIVE:NOTAM:Icao:Text
ACTIVE:RWY:ARR:ArrRwyList:DEP:DepRwyList

The active line is optional. If there is no active line, the area will not be automatically activated. An area can contain more than one active line; the plugin will check all of them to set the activation status.

The first option will activate the area automatically without any time limits when the plugin is loaded. Note that this option cannot be used together with other ACTIVE lines as it would override any other schedule.

The second option can be used to set activation schedules.

- SchedStartDate — First day to activate the area
  - month and day in the format MMDD (for recurring periods every year)
  - year, month and day in the format YYMMDD (for a single period)
- SchedEndDate — Last day to activate the area, formats as above
- SchedWeekdays — Days of the week to activate the area
  - list of numbers representing the days to activate the area, for example "145" means the area will activate on Mondays, Thursdays and Fridays
  - "0" (zero) to activate the area continuously from StartTime on SchedStartDate to EndTime on SchedEndDate
- StartTime — Time to activate the area (UTC time in the format HHMM)
- EndTime — Time to deactivate the area (UTC time in the format HHMM)

Note: SchedEndDate and SchedWeekdays only limit the activation of the area. If the activation time extends past midnight, the area stays active until EndTime on the following day.

The third option activates the area based on NOTAM information. The plugin searches for NOTAMs for the given ICAO code that contain the given text (and the text " ACT"). If those are found, the plugin attempts to set the activation schedule based on the other contents of the NOTAM.

- Icao — ICAO location indicator that publishes activation NOTAMs for the area
- Text — Text to search for in the NOTAM

The fourth option activates the area based on active runways. If all the specified runways are active, the area is activated. If even one of them is not, the area will be deactivated. The runway identifiers must be in the format "<4-letter ICAO code><runwayID>", for example "EFHK15".

- ArrRwyList — Comma-separated list of runways. Enter "*" to allow any runway.
- DepRwyList — Comma-separated list of runways. Enter "*" to allow any runway.

Bound

BOUND:C:Lat:Lon:Radius

The bound line is optional but highly recommended for areas that are circle-shaped. It increases the accuracy of the calculation while at the same time reducing the amount of calculations required, giving both an accuracy and a performance gain to the plugin.

The bound line should only be used for areas that are circles. The *"Lat"* and *"Lon"* coordinates (decimal degrees or sector file format) define the center point and the *"Radius"* (nautical miles, decimal number) the radius of the circle. The information is used to check if the aircraft is inside the area, but also the coordinate lines are still needed as they are used to draw the area on the screen (the coordinates will not be used for any calculations so make sure you only use this line for circular areas!).

Limits

LIMITS:Alt_{min}:Alt_{max}

The limits line is optional. It defines the default vertical limits of the area (in hundreds of feet). They can be changed as required in the area windows. When an area without default vertical limits is activated, its limits will be set to 0 and 999 (from 0ft to FL999). Those are also the minimum and maximum allowed values.

NoAPW

NOAPW

The optional NoAPW line inhibits APW warnings for the area.

APW_Buffer_Lat

APW_BUFFER_LAT:BufferU:BufferLI:BufferLV

The optional APW_Buffer_Lat line can be used to override the default lateral buffers applied to the area for APW processing. The first value is used above the setting value "APW_Buffer_Lat_SepLevel", the others at or below it.

- BufferU High level buffer (nautical miles, decimal value, range 0.0-999.0)
- BufferLI Low level buffer for IFR flights (nautical miles, decimal value, range 0.0-999.0)
- BufferLV Low level buffer for VFR flights (nautical miles, decimal value, range 0.0-999.0)

APW_Buffer_Vert

APW_BUFFER_VERT:BufferU:BufferLI:BufferLV

The optional APW_Buffer_Vert line can be used to override the default vertical buffers applied to the area for APW processing. The first value is used above the minimum RVSM level, the others at or below it.

- BufferU High level buffer (feet, integer value, range 0-9999)
- BufferLI Low level buffer for IFR flights (feet, integer value, range 0-9999)
- BufferLV Low level buffer for VFR flights (feet, integer value, range 0-9999)

NoSAP

NOSAP

The optional NoSAP line inhibits SAP warnings for the area.

SAP_Buffer_Lat

SAP_BUFFER_LAT:BufferU:BufferLI:BufferLV

The optional SAP_Buffer_Lat line can be used to override the default lateral buffers applied to the area for SAP processing. The first value is used above the setting value "SAP_Buffer_Lat_SepLevel", the others at or below it.

- BufferU High level buffer (nautical miles, decimal value, range 0.0-999.0)
- BufferLI Low level buffer for IFR flights (nautical miles, decimal value, range 0.0-999.0)
- BufferLV Low level buffer for VFR flights (nautical miles, decimal value, range 0.0-999.0)

SAP_Buffer_Vert

SAP_BUFFER_VERT:BufferU:BufferLI:BufferLV

The optional SAP_Buffer_Vert line can be used to override the default vertical buffers applied to the area for SAP processing. The first value is used above the minimum RVSM level, the others at or below it.

- BufferU High level buffer (feet, integer value, range 0-9999)
- BufferLI Low level buffer for IFR flights (feet, integer value, range 0-9999)
- BufferLV Low level buffer for VFR flights (feet, integer value, range 0-9999)

NoMSAW

NOMSAW

The optional NoMSAW line inhibits MSAW warnings inside the area.

Coordinate

Lat Lon

Each area definition must have at least three coordinate lines (three points being the minimum required to create a closed area, a triangle). There is practically no upper limit for the number of coordinate points, but as the required calculations increase proportionally to the number of points, it's best to keep the areas simple. The latitude and longitude values must be either in decimal degrees or in the sector file format and there must be one or more spaces between them. There may also be one or more spaces in the beginning of the line before the latitude value so it should be relatively easy to create areas from the REGIONS part of a sector file.

2.3 TopSkyAreasManualAct.txt

This file can be used to temporarily clear all defined activation rules and/or add new ones to the areas defined in the TopSkyAreas.txt file. If the file is present, the plugin will display a message box to the user as a reminder about the non-standard configuration.

To clear all existing rules, use the following syntax:

AreaName:CLEAR

- AreaName Area name as specified in TopSkyAreas.txt

Note that if you are adding new rules as well, they must be defined after this line, otherwise they'll get wiped out as well…

To add new rules, all the "ACTIVE" line types from TopSkyAreas.txt can be used. The only difference in the syntax is that in the place of the "ACTIVE" keyword, the area name must be used:

AreaName:1
AreaName:SchedStartDate:SchedEndDate:SchedWeekdays:StartTime:EndTime
AreaName:NOTAM:Icao:Text
AreaName:RWY:ARR:ArrRwyList:DEP:DepRwyList

- AreaName Area name as specified in TopSkyAreas.txt
- Other fields See TopSkyAreas.txt "ACTIVE" line types

2.4 TopSkyCPDLC.txt

This file contains data for the CPDLC system. It defines the departure clearance format(s) and pre-defined CPDLC free text items that are available and connects controller position IDs to CPDLC logins.

Departure clearance format definition

DCL:EFHK:SID:<adep2> PDC <number> . . . . . CLD <hour><min> <year2><month2><day> <adep2> PDC <number> <callsign> <endheader>CLRD TO <ades> OFF <drwy> VIA <sid> DCT <npt> SQUAWK <assr> <nfreq> <qnh> <rmk>

Type:Adep:Subtype:Text

- Type — One of the following:
  - DCL — DCL type message (other than US)
  - DCL_US — DCL type message used in USA
  - PDC — PDC type message
- Adep — Comma-separated list of departure airport ICAO codes (or "*" to match any airport)
- Subtype — One of the following:
  - AHDG — match when aircraft has an assigned heading
  - SID — match when aircraft has a SID
  - AHDG+SID — match when the aircraft has both
  - * — always match
- Text — The clearance message that is sent

Note that if both "AHDG" and "AHDG+SID" formats are defined for an airport with automatically assigned SIDs, the "AHDG" format will only be used when a SID isn't automatically assigned (an automatically assigned SID cannot be removed without changing the flight plan to prevent the automatic assignment).

As an airport typically provides either DCL or PDC (or PDC and DCL_US) type clearances, both DCL and PDC types should not be defined here for any airport. They are requested using the same message format, and if both clearance types are defined, the plugin won't know which one to send. Having both PDC and DCL_US for one airport is OK as they are requested with different messages.

The first definition line that matches both the *Adep* and *Subtype* will be used. The *Text* can contain some data fields that are automatically filled by the plugin. Some are highlighted in Project A330 DCDU, one of the choices for CPDLC software for pilots:

Data fields that are not highlighted:

- <year4> — current UTC year
- <year2> — last two digits of the current UTC year
- <month3> — current UTC month (three characters: "JAN", "FEB", etc.)
- <month2> — current UTC month (two digits)
- <day> — current UTC day (two digits)
- <hour> — hours of the current UTC time (two digits)

- <min> minutes of the current UTC time (two digits)
- <sec> seconds of the current UTC time (two digits)
- <wday3> current UTC weekday (three characters: “MON”, “TUE”, etc.)
- <number> running number of sent PDCs (three digits)
- <adep2> departure airport ICAO code

Data fields that are highlighted in DCL and DCL_US clearances:

- <callsign> callsign of the aircraft
- <adep> departure airport ICAO code
- <ades> destination airport ICAO code
- <drwy> departure runway identifier
- <ahdg> assigned heading (three digits)
- <ahdg/trk> assigned heading (“HDG xxx”) or track (“TRACK xxx”)
- <sid> SID identifier
- <npt> next route point
- <sid+npt> SID identifier, a blank space and the next route point
- <freq> controller primary frequency
- <nfreq> “NEXT FREQ ” + next frequency
- <rfreq> “REPORT READY ON ” + next frequency
- <dfreq> “DEPARTURE FREQ ” + departure frequency
- <cfl> cleared altitude/FL
- <rfl> requested altitude/FL
- <copx> exit coordination point
- <assr> assigned transponder code (four digits)
- <eobt> estimated off-blocks time (four digits)
- <aobt> actual off-blocks time (four digits)
- <aobtMDI> same as <aobt>, except displays “MDI” if equal to estimated off-blocks time
- <startup> either “START-UP APPROVED” or “REPORT READY ON ” + next frequency
- <qnh> “QNH ” + the QNH value (3-4 digits), nothing printed if unavailable
- <qnh4> same as <qnh> but always with four digits
- <qnhQ> “Q” + the QNH value (four digits), nothing printed if unavailable
- <rmk> remarks text

One special field exists. The entire message is always sent to the aircraft, but if the *Text* contains the field “<endheader>”, only the part after it will be shown in the plugin’s CPDLC message windows.

Highlighted items should have some text between them to work correctly unless the second item contains static text before the actual highlighted value. In cases where no text is desired between items, the underscore character “_” works well and shows as a blank space in the current pilot clients. Creating the clearance formats usually requires some testing to get them right.

CPDLC free text definition

FREETEXT:ReplyType:Text

- ReplyType — The reply expected from the aircraft. One of the following:
  - WU — "WILCO" or "UNABLE"
  - AN — "AFFIRM" or "NEGATIVE"
  - R — "ROGER"
  - NE — no reply expected
- Text — The message text that is sent

CPDLC login definition

LOGIN:Login:RadioCall:ControllerID

- Login — The CPDLC login to match with the controller ID
- RadioCall — The RTF callsign used by the controller
- ControllerID — The controller ID to match with the CPDLC login

As there is no way to match a controller to a CPDLC login automatically, this list is used. It follows that it's very important to use only agreed CPDLC logins for the CPDLC connection handovers to work properly.

2.5 TopSkyCallsigns.txt & TopSkyCallsignsLocal.txt

These files contain the radiotelephony callsigns to be displayed in the track labels. By default, if an identifier is found in both files, the information in the "TopSkyCallsigns.txt" file is used. The following example line shows the syntax:

AAB	Abelag Aviation	ABG	Callsign definition

The format of the files is the same as in the "ICAO_Airlines.txt" file provided with EuroScope. Only one callsign must be defined per line.

Callsign definition

ThreeLetterID<tab>**OperatorName**<tab>**Callsign**

- ThreeLetterID — Three-letter designator in the flightplan
- OperatorName — Name of the operator (not used, but must contain something)
- Callsign — Radiotelephony callsign

2.6 TopSkyMaps.txt & TopSkyMapsLocal.txt

These files contain the definitions for the maps in the Maps Window. The difference is that the maps in the "Local" file are automatically assigned to a folder named "LMAPS" and any map folder names in that file are disregarded. The following example area is used to show the syntax (optional lines in grey color):

COLORDEF:Approach:255:255:255	Color definition
SYMBOLDEF:Fix1	Symbol definition
MOVETO:-4:3	Moveto
LINETO:0:-4	Lineto
LINETO:4:3	Lineto
LINETO:-4:3	Lineto
// EFKU VOR app rwy 15	Comment
MAP:EFKU VOR 15	Name
FOLDER:IAP	Folder
COLOR:Approach	Color
LINE:N063.01.03.067:E027.47.04.942:N063.11.03.016:E027.39.18.377	Line
LINE:N063.01.22.882:E027.49.09.775:N063.00.43.220:E027.45.00.157	Line
LINE:N063.06.39.890:E027.45.04.332:N063.06.00.228:E027.40.53.958	Line
STYLE:Dot:1	Line style
LINE:N063.01.03.067:E027.47.04.942:N062.58.47.170:E027.48.49.840	Line
LINE:N062.58.47.170:E027.48.49.840:N062.54.47.000:E027.53.04.000	Line
SYMBOL:Fix1:N063.06.19.000:E027.43.04.000:ROVSU:20:-10	Symbol + label
SYMBOL:Fix1:N062.54.47.000:E027.53.04.000:VEKEM:20:-10	Symbol + label

The mandatory items for each map are a name, a folder it belongs to in the Maps Window, and a color with which to draw the items.

Color definition

COLORDEF:ColorName:R:G:B

Every color used in the maps (with some exceptions listed in the COLOR line specification) must be defined using one of these lines.

- ColorName — Color name to be used in the Color lines (text string)
- R — Color's red component value (0-255)
- G — Color's green component value (0-255)
- B — Color's blue component value (0-255)

Symbol definition

SYMBOLDEF:SymbolName

The first line for each symbol definition must be a name line.

- SymbolName Symbol name to use for this symbol in the Symbol lines (text string)

The symbol itself can consist of various elements, drawn by the following lines. The X and Y coordinates are relative to the symbol centerpoint, with the X axis having increasing values to the right and the Y axis having increasing values to the down direction. The commands are the same as in the EuroScope Symbology dialog with the exception of the possibility to draw elliptical arcs and the ":" separating the values here so the ES dialog can be used in most cases to test the results.

MOVETO:X:Y

Sets the starting point for the next LINETO command

- X Number of pixels from the symbol centerpoint in the left(-)-right(+) direction
- Y Number of pixels from the symbol centerpoint in the up(-)-down(+) direction

LINETO:X:Y

Draws a straight line from the previous position

- X Number of pixels from the symbol centerpoint in the left(-)-right(+) direction
- Y Number of pixels from the symbol centerpoint in the up(-)-down(+) direction

SETPIXEL:X:Y

Paints the selected pixel

- X Number of pixels from the symbol centerpoint in the left(-)-right(+) direction
- Y Number of pixels from the symbol centerpoint in the up(-)-down(+) direction

ARC:X:Y:Radius:StartAngle:EndAngle
ARC:X:Y:$Radius_X$:$Radius_Y$:StartAngle:EndAngle

Draws a part of a circle

- X Centerpoint offset from the symbol centerpoint in the left(-)-right(+) direction
- Y Centerpoint offset from the symbol centerpoint in the up(-)-down(+) direction
- Radius Arc radius in pixels (to make a circular arc)
- $Radius_X$ Arc radius in relation to the X axis in pixels (to make an elliptical arc)
- $Radius_Y$ Arc radius in relation to the Y axis in pixels (to make an elliptical arc)
- StartAngle Arc starting angle (integer degrees, 0 degrees is at positive X-axis, increasing counterclockwise)
- EndAngle Arc ending angle (integer degrees, 0 degrees is at positive X-axis, increasing counterclockwise)

FILLARC:X:Y:Radius:StartAngle:EndAngle
FILLARC:X:Y:$Radius_X$:$Radius_Y$:StartAngle:EndAngle

Otherwise the same as ARC above but the result is filled

POLYGON:X_1:Y_1: X_2:Y_2:…: X_n:Y_n

Draws a filled polygon with n vertices

Name

MAP:MapName

The first line for each map definition must be a name line. This will identify the map in the Maps Window.

- MapName Map name to identify it in the Maps window (text string)

Folder

FOLDER:FolderName

Every map must belong to a folder. There is practically no limit to how many maps a single folder can contain. The following folder names get special treatment: “ARTCC HIGH”, “ARTCC”, “ARTCC LOW”, “GEO”, “SID”, “STAR” and “FREE TEXT”. Maps assigned to those folders will not be shown in the Maps Window. Instead, if sector file items with the same names are found in the active sector file (i.e. a SID named ABC1A exists in the sector file and a map named “ABC1A” in folder “SID”), the map is assigned as additional data for that sector file item and activated/deactivated together with it.

- FolderName Folder name to identify it in the Maps window (text string)

Note: the folder name may not start with an empty space character, and may not contain the backslash (“\”) character.

Zoom

ZOOM:ZoomLevel

A whole map or parts of it can be hidden based on the current zoom level. When the radar screen is zoomed out so that there are less than the specified number of pixels per nautical mile, the lines of the map definition after this line are not read. There can be more than one zoom line in one map to hide parts of the map at different zoom levels.

- ZoomLevel Radar screen zoom level (pixels per nautical mile, decimal value)

Note: when there is more than one zoom line in a map, their order is important (for example "ZOOM:5" has to be before "ZOOM:10" to have any effect as with zoom below 10 pix/nm the "ZOOM:5" line will never be read if it's after the "ZOOM:10" line...)

Active

ACTIVE:1
ACTIVE:SchedStartDate:SchedEndDate:SchedWeekdays:StartTime:EndTime
ACTIVE:NOTAM:Icao:Text
ACTIVE:RWY:ARR:ArrRwyList:DEP:DepRwyList

The active line is optional. If there is no active line, the map will not be automatically activated. A map can contain more than one active line; the plugin will check all of them to set the activation status. The map will be placed in the "AUTO" folder, a folder name cannot be specified for automatically activating maps.

The first option will activate the map automatically without any time limits when the plugin is loaded. Note that this option cannot be used together with other ACTIVE lines as it would override any other schedule.

The second option can be used to set activation schedules.

- SchedStartDate — First day to activate the map
  - month and day in the format MMDD (for recurring periods every year)
  - year, month and day in the format YYMMDD (for a single period)
- SchedEndDate — Last day to activate the map, formats as above
- SchedWeekdays — Days of the week to activate the map
  - list of numbers representing the days to activate the map, for example "145" means the map will activate on Mondays, Thursdays and Fridays
  - "0" (zero) to activate the map continuously from StartTime on SchedStartDate to EndTime on SchedEndDate
- StartTime — Time to activate the map (UTC time in the format HHMM)
- EndTime — Time to deactivate the map (UTC time in the format HHMM)

Note: SchedEndDate and SchedWeekdays only limit the activation of the map. If the activation time extends past midnight, the map stays active until EndTime on the following day.

The third option activates the map based on NOTAM information. The plugin searches for NOTAMs for the given ICAO code that contain the given text (and the text " ACT"). If those are found, the plugin attempts to set the activation schedule based on the other contents of the NOTAM.

- Icao — ICAO location indicator that publishes activation NOTAMs for the map
- Text — Text to search for in the NOTAM

The fourth option activates the map based on active runways. If all the specified runways are active, the map is activated. If even one of them is not, the map will be deactivated. The runway identifiers must be in the format "<4-letter ICAO code><runwayID>", for example "EFHK15".

- ArrRwyList — Comma-separated list of runways. Enter "*" to allow any runway.
- DepRwyList — Comma-separated list of runways. Enter "*" to allow any runway.

Color

COLOR:ColorName
COLOR:ColorName:FillColorName

Every map must have at least one color line. It sets the color to be used to draw the subsequent drawings. Each line, symbol, etc. within a map can be drawn with a different color simply by including a new color line when a color change is required. If the FillColorName is not specified, it is set to the same color as ColorName. All used color names (with some exceptions listed below) must be defined in the file using a COLORDEF line.

- ColorName Color to be used for drawing lines and texts
- FillColorName Color to be used for filling the FILLARC and POLYGON objects

Some plugin colors can be used as map colors without having to define them in a COLORDEF line (however if the color is defined in a COLORDEF line, that definition will override the plugin color definition). The available color names are:

- Active_Map
- Active_Map_Type_1 … Active_Map_Type_20
- Active_RD_Map
- Active_Text_Map
- Map_1 … Map_4
- Map_Hotspot
- Map_Info
- Preactive_Map
- Preactive_Text_Map
- Predisplay_Map
- Rwy_App_Line_Inuse
- Rwy_App_Line_Not_Inuse

Style

STYLE:StyleName
STYLE:StyleName:Width

The style line defines the line type for any subsequent LINE items within this map. It is not mandatory, a Solid type line with width 1 pixel will be drawn by default. As with the COLOR line, a single map may contain any required number of STYLE lines to draw different line styles within the same map. The width is only necessary for Solid type lines (it has no effect on the other types). If a width is not defined, a 1 pixel width is drawn by default.

- StyleName Style to be used (Solid, Dash, Dot, DashDot or DashDotDot)
- Width Width of line (pixels)

Line

LINE:Lat$_1$:Lon$_1$:Lat$_2$:Lon$_2$
LINE:StartPointName:EndPointName

Draws a line from one point to another. Uses the previously defined line style (or solid line with 1 pixel width if no style defined).

- Lat_1 Latitude of starting point (decimal degrees or sector file format)
- Lon_1 Longitude of starting point (decimal degrees or sector file format)
- Lat_2 Latitude of end point (decimal degrees or sector file format)
- Lon_2 Longitude of end point (decimal degrees or sector file format)
- StartPointName Fix, VOR, NDB ,airport code or runway (must be found in the active sector file)
- EndPointName Fix, VOR, NDB, airport code or runway (must be found in the active sector file)

Note: the syntax to define a runway threshold as a StartPointName or an EndPointName is the 4-letter ICAO airport designator followed by a forward slash and the runway identifier.

Symbol

SYMBOL:SymbolName:Lat:Lon
SYMBOL:SymbolName:Lat:Lon:Label:OffsetX:OffsetY
SYMBOL:SymbolName:PointName
SYMBOL:SymbolName:PointName:Label:OffsetX:OffsetY

Draws a predefined symbol on the screen. Optionally, can display a text label with the label's centerpoint offset a given number of pixels from the symbol centerpoint.

- SymbolName Name of symbol
- Lat Latitude of symbol centerpoint (decimal degrees or sector file format)
- Lon Longitude of symbol centerpoint (decimal degrees or sector file format)
- PointName Fix, VOR, NDB, airport code or runway (must be found in the active sector file)
- Label Text label (text string)
- OffsetX Number of pixels to offset the label centerpoint in the left(-)-right(+) direction
- OffsetY Number of pixels to offset the label centerpoint in the up(-)-down(+) direction

Note: the syntax to define a runway threshold as a PointName is the 4-letter ICAO airport designator followed by a forward slash and the runway identifier.

Text

TEXT:Lat:Lon:Label
TEXT:Lat:Lon:Label:OffsetX:OffsetY
TEXT:PointName:Label
TEXT:PointName:Label:OffsetX:OffsetY

Draws a text label on the screen. Optionally, the label's centerpoint can be offset a given number of pixels from the given position.

- Lat — Latitude of label centerpoint (decimal degrees or sector file format)
- Lon — Longitude of label centerpoint (decimal degrees or sector file format)
- PointName — Fix, VOR, NDB, airport code or runway (must be found in the active sector file)
- Label — Text label (text string)
- OffsetX — Number of pixels to offset the label centerpoint in the left(-)-right(+) direction
- OffsetY — Number of pixels to offset the label centerpoint in the up(-)-down(+) direction

Note: the syntax to define a runway threshold as a PointName is the 4-letter ICAO airport designator followed by a forward slash and the runway identifier.

2.7 TopSkyMSAW.txt

The file is read one line at a time and the first line that contains the aircraft position returns the minimum safe altitude, so put specific small area lines at the top and large general areas to the end of the file. Be careful to always use the correct syntax to achieve any results and use the point (.) as the decimal separator. For backward compatibility, also the comma (,) is accepted to separate the fields. Only define one area per line. There are five types of area definitions that are accepted:

Lat/Lon box area

A:Lat_{min}:Lat_{max}:Lon_{min}:Lon_{max}:MSA

An area bounded by the minimum and maximum latitude and longitude values

- Lat_{min} Minimum Latitude (decimal degrees or sector file format)
- Lat_{max} Maximum Latitude (decimal degrees or sector file format)
- Lon_{min} Minimum Longitude (decimal degrees or sector file format)
- Lon_{max} Maximum Longitude (decimal degrees or sector file format)
- MSA Minimum Safe Altitude within the area (feet, integer value)

Circle

C:Lat:Lon:R:MSA

A circle of radius r with center point at (Lat,Lon)

- Lat Circle center point latitude (decimal degrees or sector file format)
- Lon Circle center point longitude (decimal degrees or sector file format)
- R Radius of circle (nautical miles, decimal value)
- MSA Minimum Safe Altitude within the circle (feet, integer value)

Lat/Lon box area list

L:Lat_{min}:Lon_{min}:ΔLat:ΔLon:N:MSA_1:MSA_2:…:MSA_n

A series of latitude-longitude bounded boxes. The boxes are in an east-west direction, with the first box being the westernmost.

- Lat_{min} Latitude of the south edge of the boxes (decimal degrees or sector file format)
- Lon_{min} Longitude of the west edge of first box (decimal degrees or sector file format)
- ΔLat Latitude size of one box (decimal degrees)
- ΔLon Longitude size of one box (decimal degrees)
- N Number of boxes
- MSA_1-MSA_n Minimum Safe Altitudes of the boxes (feet, integer values, must be n values)

Polygon

P:N:Lat_1:Lon_1:Lat_2:Lon_2:…:Lat_n:Lon_n:MSA

A polygon with *n* vertices at given latitude-longitude points

- N — Number of vertices
- Lat_1 — Latitude of vertex 1 (decimal degrees or sector file format)
- Lon_1 — Longitude of vertex 1 (decimal degrees or sector file format)
- Lat_2 — Latitude of vertex 2 (decimal degrees or sector file format)
- Lon_2 — Longitude of vertex 2 (decimal degrees or sector file format)
- Lat_n — Latitude of vertex n (decimal degrees or sector file format)
- Lon_n — Longitude of vertex n (decimal degrees or sector file format)
- MSA — Minimum Safe Altitude within the polygon (feet, integer value)

Sector

S:Lat:Lon:$TRdl_1$:$TRdl_2$:R_{min}:R_{max}:MSA

An area defined as being between two true bearings from a point (Lat,Lon) - clockwise direction from Rdl_1 to Rdl_2 - and between distances R_{min} and R_{max} from the point

- Lat — Point latitude (decimal degrees or sector file format)
- Lon — Point longitude (decimal degrees or sector file format)
- $TRdl_1$ — Bearing 1 (degrees true, decimal value)
- $TRdl_2$ — Bearing 2 (degrees true, decimal value)
- R_{min} — Minimum distance from point (nautical miles, decimal value)
- R_{max} — Maximum distance from point (nautical miles, decimal value)
- MSA — Minimum Safe Altitude within the sector (feet, integer value)

2.8 TopSkyRadars.txt

The file contains the primary radar station definitions to be used for displaying raw video radar data. The following example shows the syntax (optional lines in grey color):

// Helsinki PSR	Comment
RADAR:Helsinki	Radar
POSITIONS:EFHF:EFHK:EFES	Positions
LOCATION:N060.18.56.400:E024.57.54.400	Location
ALTITUDE:335	Altitude
BEAMWIDTH:1.4	Beamwidth
PULSEWIDTH:1.0	Pulsewidth
MAXANGLE:50	Maxangle
RANGE:0.0:60.0	Range

Radar

RADAR:RadarName

Each radar definition must start with a Radar line that defines the radar station name.

- RadarName Radar station name (text string)

Positions

POSITIONS:Pos_1:Pos_2:…

Defines the list of controller positions that use the radar station. Only one radar station can be active so the first station in the file that contains a match will be used. The logic compares the positions against the login callsign from the beginning of the string (“EF” will be a match for either “EFES_CTR” or “EFHK_TWR” but not for “SAEF_APP”)

- Pos_x Position login callsign (text string, full callsign or first x letters)

Location

LOCATION:Lat:Lon

The location of the radar antenna.

- Lat Latitude value (decimal degrees or sector file format)
- Lon Longitude value (decimal degrees or sector file format)

Altitude

ALTITUDE:Alt

The radar antenna altitude above mean sea level. If not specified, a value of 0 is used.

- Alt The antenna altitude (feet AMSL, integer value)

Beamwidth

BEAMWIDTH:Beamwidth

Specifies the beamwidth of the radar in degrees. If not specified, a value of 1.5 will be used. The value affects how wide the radar targets will be (twice the beamwidth value).

- Beamwidth The beamwidth of the radar (degrees, decimal value)

Pulsewidth

PULSEWIDTH:Pulsewidth

Specifies the pulsewidth of the radar in microseconds. If not specified, a value of 1.0 will be used. The value affects how deep the radar targets will be (approx. 0.08nm/microsecond).

- Pulsewidth The pulsewidth of the radar (microseconds, decimal value)

Maxangle

MAXANGLE:MaxAngle

Defines the maximum elevation angle of the radar measured from horizontal level. If not specified, a value of 90 will be used (i.e. coverage all the way up to vertical)

- MaxAngle The maximum vertical angle of the radar (degrees, integer value)

Range

RANGE:MinRange:MaxRange

Defines the minimum and maximum detection ranges of the radar station. If not specified, a minimum value of 0 and a maximum value of 999999 will be used.

- MinRange The minimum detection range of the radar (nautical miles, decimal value)
- MaxRange The maximum detection range of the radar (nautical miles, decimal value)

Ceiling

CEILING:Ceiling

Defines the maximum detection altitude of the radar. If not specified, a value of 999999 will be used.

- Ceiling The maximum detection altitude of the radar (feet AMSL, integer value)

2.9 TopSkySettings.txt & TopSkySettingsLocal.txt

These two files allow changing the plugin settings. The difference between them is that the settings in the first file are loaded every time, while the settings in the "Local" file are only loaded on user request (by performing a "Sign In"). The settings in the "Local" file can be either general or login callsign specific.

General settings have to be located at the beginning of the "Local" file before any login callsign specific ones. Login callsign specific ones are defined by creating sections starting with a line that contains a text string in square brackets above them.

The login callsign specific settings are checked by comparing the text string against the login callsign. If the login callsign contains the text, any settings after that are loaded until a new line with text in square brackets is found which after the check is done again. If not, all the settings in that section are skipped.

Example TopSkySettingsLocal.txt file:

Setting1=0	
Setting2=123	
[_CTR]	New section start
Setting2=100	
Setting3=0	
[EFES_]	New section start
Setting2=200	
[ESSA_TWR]	New section start
Setting2=300	
Setting4=0	

When performing a Sign In, first "Setting1" is set to "0" and "Setting2" to "123" regardless of the login callsign. If the login callsign is for example "EFES_2_CTR", the first section ("_CTR") is a match, so "Setting2" will be changed to "100" and "Setting3" is set to "0". The second section ("EFES_") is also a match, so "Setting2" is changed once again, this time to "200". The last section ("ESSA_TWR") is not a match so the settings there won't be applied. So, any one setting can appear in the file more than once, and be set more than once depending on how the file is laid out. As the file is always read in the order it is written, the more specific sections should be at the bottom (like in the example, any _CTR callsign will get a different "Setting2" value than the other callsigns, but if it happens to be EFES_CTR or EFES_<anything here>_CTR, the value is different from the other _CTR callsigns).

The available settings, their default values and acceptable ranges are described in an Excel spreadsheet provided together with this file.

2.10 TopSkySSRcodes.txt

The file contains the SSR code range and area definitions to be used when assigning transponder codes via the plugin. The following example shows the syntax (optional lines in grey color):

AREA:EF1	Area
RADIUS:15.0	Radius
N060.00.00.000 E025.00.00.000	Coordinate
// domestic secondary	Comment
RANGE:3201:3277	Range
ADES:EF	ADES
PROTECTION:EF1	Protection
SECONDARY	Secondary
IFR	IFR

The plugin's SSR code assignment system checks the flightplan, finds out which code ranges are available for that flightplan, checks for codes already in use, and then assigns one of the available codes. Transponder codes ending with "00" are not assigned.

The example area above designates the code range 3201-3277 available for IFR traffic with destinations starting with "EF" and not entering the area "EF1". In addition, the range is defined to be a secondary one, to be used only if there are no available codes found in other ranges.

Mode S global

MODE_S_GLOBAL

If this line appears anywhere in the file, it indicates that the identity of mode S capable aircraft will be maintained using mode S information everywhere. See also "Mode S" line to limit the availability to specified area(s).

Area

AREA:AreaName

Each area definition must start with an Area line that defines the area name. The area must be defined in the file before it is referred to in a code range definition. The area names are case sensitive.

- AreaName Area name to use in the code assignment rules (text string)

Radius

RADIUS:Radius

If the area is a circle, it can be defined as a center point and a distance from it. In this case the area definition needs the radius line and one coordinate line (see below). All other area shapes need to be defined as polygons using three or more coordinate lines, and then the radius line shall not be used.

- Radius Area radius (nautical miles, decimal value)

Mode S

MODE_S

This line indicates that inside this area the identity of mode S capable aircraft will be maintained using mode S information. The mode S conspicuity code (1000) can be assigned by the plugin when the entire planned route is within these areas.

Coordinate

Lat Lon

Each area definition must have either at least three coordinate lines (three points being the minimum required to create a closed area, a triangle), or one coordinate line and a radius line (see above). There is practically no upper limit for the number of coordinate points, but as the required calculations increase proportionally to the number of points, it's best to keep the areas simple. The latitude and longitude values must be either in decimal degrees or in the sector file format and there must be one or more spaces between them. There may also be one or more spaces in the beginning of the line before the latitude value.

Range

RANGE:StartCode:EndCode

This line is the only mandatory line for a code range definition and must always be the first line in a definition. It starts the definition by specifying the range of codes in it.

- StartCode The first code in the range (4 octal digits, 0001-7777)
- EndCode The last code in the range (4 octal digits, 0001-7777)

Adhoc

ADHOC

This line causes the codes in this range to be assigned only to flightplans where the departure and/or destination fields are empty.

Mandatory

MANDATORY

This line causes the codes in this range to be mandatory for all matching flights. Possible other matching, but non-mandatory code ranges will be considered only if there are no assignable codes found in the mandatory ranges.

Secondary

SECONDARY

This line defines the range to be a secondary one, to be used only when the other ranges do not have any assignable codes left.

IFR

IFR

This line causes the codes in this range to be assigned only to IFR flightplans.

VFR

VFR

This line causes the codes in this range to be assigned only to VFR flightplans.

Direction

DIRECTION:TTrk1:TTrk2

This line limits the code assignment to flights having a track between the two specified true tracks (clockwise direction from Trk1 to Trk2). If VIA and/or NOTVIA lines are also present in the code range, the track to be checked is the outbound track from the specified point(s). If not, the tracks are checked against the aircraft's planned track from its present position.

- TTrk1 Start angle for the track range (degrees true, decimal value)
- TTrk2 End angle for the track range (degrees true, decimal value)

ADEP

ADEP:ICAOcode:ICAOcode:ICAOcode:…

This line limits the code assignment to flights departing from one of the defined airports. The whole ICAO airport code is not needed; the match can also be done on the first one or more letters, e.g. entering "EF" will match all airports with ICAO designators beginning with "EF". The ADEP line can contain one or more airport codes and one code range definition can also have more than one ADEP line if necessary.

- ICAOcode Airport ICAO code (complete or partial)

NotADEP

NOTADEP:ICAOcode:ICAOcode:ICAOcode:…

This line limits the code assignment to flights not departing from any of the defined airports.Otherwise the format and limitations are the same as in the ADEP line.

- ICAOcode Airport ICAO code (complete or partial)

ADES

ADES:ICAOcode:ICAOcode:ICAOcode:…

This line limits the code assignment to flights arriving at one of the defined airports.Otherwise the format and limitations are the same as in the ADEP line.

- ICAOcode Airport ICAO code (complete or partial)

NotADES

NOTADES:ICAOcode:ICAOcode:ICAOcode:…

This line limits the code assignment to flights not arriving at any of the defined airports.Otherwise the format and limitations are the same as in the ADEP line.

- ICAOcode Airport ICAO code (complete or partial)

Local

LOCAL:ICAOcode:ICAOcode:ICAOcode:…

This line limits the code assignment to local flights (ADEP=ADES) from one of the defined airports. The whole ICAO airport code is not needed; the match can also be done on the first one or more letters, e.g. entering “EF” will match all airports with ICAO designators beginning with “EF”. The LOCAL line can contain one or more airport codes and one code range definition can also have more than one LOCAL line if necessary. The LOCAL line cannot be used together with ADEP or ADES lines.

- ICAOcode Airport ICAO code (complete or partial)

Via

VIA:Point:Point:Point:…

This line limits the code assignment to flights routing via at least one of the defined points. The point can be anywhere along the flightplan. One or more points can be defined in one VIA line and one code range definition can contain more than one VIA line if necessary.

- Point Point name (Fix, VOR, NDB or airport)

NotVia

NOTVIA:Point:Point:Point:…

This line limits the code assignment to flights not routing via any of the defined points. Otherwise the format and limitations are the same as in the VIA line.

- Point Point name (Fix, VOR, NDB or airport)

AreaVia

AREAVIA:AreaName:AreaName:AreaName:…

This line limits the code assignment to flights routing via at least one of the defined areas. The area(s) must have been defined earlier in the data file. One or more areas can be defined in one AREAVIA line and one code range definition can contain more than one AREAVIA line if necessary.

- AreaName Area name (text string)

Protection

PROTECTION:AreaName:AreaName:AreaName:…

This line limits the code assignment to flights not routing via any of the defined areas. Otherwise the format and limitations are the same as in the AREAVIA line.

- AreaName Area name (text string)

AdepArea

ADEPAREA:AreaName:AreaName:AreaName:…

This line limits the code assignment to departing from one of the defined areas. The area(s) must have been defined earlier in the data file. One or more areas can be defined in one ADEPAREA line and one code range definition can contain more than one ADEPAREA line if necessary.

- AreaName Area name (text string)

NotAdepArea

NOTADEPAREA:AreaName:AreaName:AreaName:…

This line limits the code assignment to flights not departing from any of the defined areas. Otherwise the format and limitations are the same as in the ADEPAREA line.

- AreaName Area name (text string)

AdesArea

ADESAREA:AreaName:AreaName:AreaName:…

This line limits the code assignment to flights arriving in one of the defined areas. The area(s) must have been defined earlier in the data file. One or more areas can be defined in one ADESAREA line and one code range definition can contain more than one ADESAREA line if necessary.

- AreaName Area name (text string)

NotAdesArea

NOTADESAREA:AreaName:AreaName:AreaName:…

This line limits the code assignment to flights not arriving in any of the defined areas. Otherwise the format and limitations are the same as in the ADESAREA line.

- AreaName Area name (text string)

Unit

UNIT:LoginCallsign:LoginCallsign:LoginCallsign:…

This line limits the code assignment based on your network login callsign. The whole callsign is not needed; the match can also be done on the first one or more letters, e.g. entering “EFES” will match callsigns beginning with “EFES”. The UNIT line can contain one or more callsigns and one code range definition can also have more than one UNIT line if necessary.

- LoginCallsign Current login callsign (complete or partial)

NotUnit

NOTUNIT:LoginCallsign:LoginCallsign:LoginCallsign:…

This line limits the code assignment to network login callsigns other than the specified ones. Otherwise the format and limitations are the same as in the UNIT line.

- LoginCallsign Current login callsign (complete or partial)

Callsign

CALLSIGN:Callsign:Callsign:Callsign:…

This line limits the code assignment based on the aircraft's callsign. The whole callsign is not needed; the match can also be done on the first one or more letters, e.g. entering "FIN" will match callsigns beginning with "FIN". The CALLSIGN line can contain one or more callsigns and one code range definition can also have more than one CALLSIGN line if necessary.

- Callsign Aircraft callsign (complete or partial)

NotCallsign

NOTCALLSIGN:Callsign:Callsign:Callsign:…

This line limits the code assignment to aircraft callsigns other than the specified ones. Otherwise the format and limitations are the same as in the CALLSIGN line.

- Callsign Aircraft callsign (complete or partial)

2.11 TopSkySTCA.txt

The file contains the definitions for final approach areas where smaller (usually 2.5nm) separation can be used, as well as STCA exclusion areas needed for parallel approaches and departures. The following example is used to show the syntax (optional lines in grey color):

FINALAPP:EFHK:04L	FinalApp
// EFHK 04's	Comment
SOIR:EFHK:04L:04R	SOIR
NOZ1:N060.18.37.790:E024.54.30.240	NOZ 1
NOZ1:N060.19.21.880:E024.53.08.680	NOZ 1
NOZ1:N060.12.36.560:E024.38.23.030	NOZ 1
NOZ1:N060.11.52.460:E024.39.44.310	NOZ 1
NOZ2:N060.18.49.470:E024.55.54.640	NOZ 2
NOZ2:N060.18.05.350:E024.57.16.110	NOZ 2
NOZ2:N060.11.20.250:E024.42.29.640	NOZ 2
NOZ2:N060.12.04.370:E024.41.08.450	NOZ 2

FinalApp

FINALAPP:AirportICAO:RwyID:Range:XTE:Lat:Lon:CourseT

This line creates a final approach area where smaller separation values for STCA are used.

- AirportICAO — The airport's ICAO code
- RwyID — The runway identifier
- Range — Maximum range from the approach end point (decimal value, range 0.0-99.0)
- XTE — Maximum cross-track error from the approach course (decimal value, range 0.0-99.0)
- Lat — Approach end point latitude (decimal degrees or sector file format)
- Lon — Approach end point longitude (decimal degrees or sector file format)
- CourseT — Approach course (degrees true, decimal value)

Only the ICAO code and RwyID are mandatory to be defined, so for example "FINALAPP:EFHK:04L" creates an approach area for that runway with default values. The default values for the other items are:

- Range — 10.0 nm
- XTE — 0.5 nm
- Lat, Lon — Runway threshold coordinates from the active sector file
- CourseT — Runway true bearing calculated from the threshold coordinates

Note that if one or more of the optional values needs to be changed, the line must include all the values up to the last changed value.

An STCA alert between two aircraft will be inhibited when all of the following are true:

- Both aircraft are planned to land on the runway in question
- Both aircraft are inside the same approach area
- Both aircraft have ground tracks within 10 degrees of the approach course
- The aircraft closer to the runway is not wake turbulence category HEAVY or SUPER, or a B757
- The aircraft further from the runway is in the same or higher wake turbulence category

SOIR

SOIR:AirportICAO:RwyID1:RwyID2:Range:WidthIn:WidthOut

This line starts the definition.

- AirportICAO The airport's ICAO code
- RwyID1 Identifier of the runway on the left side in the direction of flight
- RwyID2 Identifier of the runway on the right side in the direction of flight
- Range Length of the NOZ (nm, decimal value, range 0.0-99.0)
- WidthIn Width of the NOZ toward the other runway (nm, decimal value, range 0.0-99.0)
- WidthOut Width of the NOZ away from the other runway (nm, decimal value, range 0.0-99.0)

Only the ICAO code and runway identifiers are mandatory to be defined. The default values for the other items are:

- Range 10.0 nm
- WidthIn / WidthOut 0.3 nm

Note that if one or more of the optional values needs to be changed, the line must include all the values up to the last changed value. A custom polygon can also be defined as the NOZ, see the NOZ1 and NOZ2 lines.

For arrivals, when one aircraft is within one NOZ and another within the other, and both aircraft have ground tracks within 10 degrees of the approach tracks for their runways, an STCA warning between those aircraft will be inhibited. For departures, the conditions are the same except that the 10 degree track error limit is only valid toward the other runway. Both runways also need to be set as active for arrival (for an arrival setup) or departure (for a departure setup) in the EuroScope runway setup. Both arrival and departure setups can be active simultaneously.

If the airport has more than two runways where simultaneous approaches are performed, each possible pair must be defined separately. For example if an airport can do simultaneous approaches on any combination of runways 01L, 01C and 01R, three definitions are required (one for 01L/01C, one for 01L/01R and one for 01C/01R).

Departure

DEPARTURE

This optional line defines the setup as a departure setup. If not present, the setup is an arrival setup.

Approach course 1 and 2

CRS1:CourseT
CRS2:CourseT

This optional line is used to define the approach course if it is different from the runway bearing. For it to have any effect, the setup must be an arrival setup and the NOZ for the corresponding runway must also be defined manually (see below).

- CourseT The approach course for the runway (degrees true, decimal value)

NOZ 1 and 2

NOZ1:Lat:Lon
NOZ2:Lat:Lon

These optional lines are used to define vertices for custom polygons to override the default NOZ areas for the runways.

- Lat Point latitude (decimal degrees or sector file format)
- Lon Point longitude (decimal degrees or sector file format)

2.12 TopSkySymbols.txt

This file makes it possible to change the default symbols drawn by the plugin. The following example symbol shows the syntax:

// distance marker	Comment
SYMBOL:MARKER	Type
ELLIPSE:0:0:2	Definition

Symbol definition

SYMBOL:SymbolType

The first line for each symbol definition must be a type line.

- SymbolType — Symbol type (one of the following):
  - AIRPORT — Airport symbol
  - MARKER — Runway approach line distance marker symbol
  - RPSH — History dot symbol
  - RPSC — Coasting track symbol
  - RPS1 — Radar position symbol for PSR tracks
  - RPS2 — Radar position symbol for SSR(A) tracks (Uncontrolled SSR and SSR+PSR tracks in COOPANS version)
  - RPS2I — Radar position symbol for SSR(A) tracks with ident (Uncontrolled SSR and SSR+PSR tracks in COOPANS version)
  - RPS3 — Radar position symbol for PSR+SSR(A) tracks (Controlled SSR and SSR+PSR tracks in COOPANS version)
  - RPS3I — Radar position symbol for PSR+SSR(A) tracks with ident (Controlled SSR and SSR+PSR tracks in COOPANS version)

  The following are not used in the COOPANS version of the plugin:

  - RPS4 — Radar position symbol for SSR(S) tracks
  - RPS4I — Radar position symbol for SSR(S) tracks with ident
  - RPS5 — Radar position symbol for PSR+SSR(A) tracks
  - RPS5I — Radar position symbol for PSR+SSR(A) tracks with ident
  - RPS6 — Radar position symbol for SSR(A+S) tracks
  - RPS6I — Radar position symbol for SSR(A+S) tracks with ident
  - RPS7 — Radar position symbol for SSR(A) tracks
  - RPS7I — Radar position symbol for PSR+SSR(A+S) tracks with ident

The symbol itself can consist of various elements, drawn by the following lines. The X and Y coordinates are relative to the symbol centerpoint, with the X axis having increasing values to the right and the Y axis having increasing values to the down direction. The commands are the same as in the EuroScope Symbology dialog with the exception of the possibility to draw elliptical arcs and the ":" separating the values here so the ES dialog can be used in most cases to test the results.

MOVETO:X:Y

Sets the starting point for the next LINETO command

- X Number of pixels from the symbol centerpoint in the left(-)-right(+) direction
- Y Number of pixels from the symbol centerpoint in the up(-)-down(+) direction

LINETO:X:Y

Draws a straight line from the previous position

- X Number of pixels from the symbol centerpoint in the left(-)-right(+) direction
- Y Number of pixels from the symbol centerpoint in the up(-)-down(+) direction

SETPIXEL:X:Y

Paints the selected pixel

- X Number of pixels from the symbol centerpoint in the left(-)-right(+) direction
- Y Number of pixels from the symbol centerpoint in the up(-)-down(+) direction

ARC:X:Y:Radius:StartAngle:EndAngle
ARC:X:Y:$Radius_X$:$Radius_Y$:StartAngle:EndAngle

Draws a part of a circle or ellipse

- X Centerpoint offset from the symbol centerpoint in the left(-)-right(+) direction
- Y Centerpoint offset from the symbol centerpoint in the up(-)-down(+) direction
- Radius Arc radius in pixels (to make a circular arc)
- $Radius_X$ Arc radius in relation to the X axis in pixels (to make an elliptical arc)
- $Radius_Y$ Arc radius in relation to the Y axis in pixels (to make an elliptical arc)
- StartAngle Arc starting angle (integer degrees, 0 degrees is at positive X-axis, increasing counterclockwise)
- EndAngle Arc ending angle (integer degrees, 0 degrees is at positive X-axis, increasing counterclockwise)

FILLARC:X:Y:Radius:StartAngle:EndAngle
FILLARC:X:Y:$Radius_X$:$Radius_Y$:StartAngle:EndAngle

Otherwise the same as ARC, but the result is filled

ELLIPSE:X:Y:Radius
ELLIPSE:X:Y:$Radius_X$:$Radius_Y$

Otherwise the same as FILLARC, but always draws a complete circle or ellipse

POLYGON:X_1:Y_1: X_2:Y_2:…: X_n:Y_n

Draws a filled polygon with n vertices

2.13 TopSkyViews.txt

The file contains the definitions for the items in the View Window. Two types of definitions are allowed, enter only one definition per line:

Lat/Lon box

VIEW:ViewName:Lat_{min}:Lon_{min}:Lat_{max}:Lon_{max}

An area bounded by the minimum and maximum latitude and longitude values. The resulting screen area covers at least the required coordinates, possibly more depending on the screen shape.

- ViewName — A name to identify the view in the View Window
- Lat_{min} — Minimum Latitude (decimal degrees or sector file format)
- Lon_{min} — Minimum Longitude (decimal degrees or sector file format)
- Lat_{max} — Maximum Latitude (decimal degrees or sector file format)
- Lon_{max} — Maximum Longitude (decimal degrees or sector file format)

Centerpoint and range

VIEW:ViewName:Lat:Lon:Range
VIEW:ViewName:PointName:Range

An area defined by a range from a centerpoint. The resulting screen area will be centered on the required centerpoint and will show at least the required distance to every direction from the centerpoint.

- ViewName — A name to identify the view in the View Window
- Lat — Centerpoint Latitude (decimal degrees or sector file format)
- Lon — Centerpoint Longitude (decimal degrees or sector file format)
- PointName — Fix, VOR, NDB, airport code or runway (must be found in the active sector file)
- Range — The displayed range (nautical miles, decimal value)

Note: the syntax to define a runway threshold as a PointName is the 4-letter ICAO airport designator followed by a forward slash and the runway identifier.