

# EUROPEAN ORGANISATION FOR THE SAFETY OF AIR NAVIGATION



## **SAAM Reference Manual 4.2.0 Beta**

Saam Version	: 4.2.0 Beta
Edition Date	: 21-12-2012
Status	: Baseline
Intended for	: Restricted audience

# SAAM Reference Manual

© 1992 - 2012 EUROCONTROL

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: December 2012

## **Publisher**

*EUROCONTROL*

## **Managing Editor**

*EUROCONTROL*

## **Technical Editors**

*SAAM developers*

*Annika Lindberg*

## **Team Coordinator**

*SAAM teams*

## **Production**

*EUROCONTROL*

## **Special thanks to:**

*All the people who contributed to this document, to our Help Writer Annika, to the graphic artist who created this great product logo on the cover page.*

# Table of Contents

	0
<b>Chapter 1 Introduction to SAAM</b>	<b>10</b>
1.1 What's new .....	10
1.1.1 SAAM V4.2.0 December 2012 .....	10
1.1.2 SAAM V4.1.0 November 2012 .....	12
1.2 SAAM Requirements .....	13
1.3 Disclaimer .....	14
1.4 User Support .....	15
1.5 SAAM Installation .....	16
1.6 SAAM Data files .....	17
1.7 The SAAM Window .....	19
1.7.1 Manipulating the Picture .....	21
1.7.2 SAAM Working Directory .....	22
<b>Chapter 2 SAAM Menu Bar</b>	<b>24</b>
2.1 File Menu .....	24
2.2 Edit Menu .....	27
2.3 Geo Tools Menu .....	27
2.4 Processing Menu .....	28
2.4.1 Assignment .....	28
Network Parameters .....	32
Free route network generation .....	35
Route Option Generator .....	39
2.4.2 Profile .....	42
Check flight level constraints .....	45
2.4.3 Airspace/Traffic Intersection .....	46
2.4.4 Free Route .....	49
2.4.5 Forecast .....	51
2.4.6 Optimisation .....	52
2.4.7 Route Choice Based on Military Opening Time .....	52
2.4.8 Cheapest Route .....	54
2.4.9 SIM .....	56
2.4.10 Generic Launch .....	58
2.4.11 SAAM Bash Console .....	60
2.5 Analysis Menu .....	66
2.5.1 Queries .....	66
Query Dialog Box .....	67
Clean T5 .....	70
Generating a sub-SO6 File .....	71
Information Line .....	72
Search Criteria Suffix .....	73
The Criteria Browser .....	75
Multi-Criteria Query or QIF .....	76
2.5.2 3D Densities .....	78
2.5.3 Conflict .....	81
2.5.4 Surface / Volume .....	85
2.5.5 Route Length .....	85
2.5.6 Maximum FL .....	89

2.5.7	Airspace Load .....	90
	Advanced Parameters Dialog Box.....	92
	Entry Rate as calculated by CFMU and NEVAK.....	93
	Sliding Hourly Entry Rate (SHER).....	94
	Overload .....	96
	View ing the Results.....	97
2.5.8	Combine Airspace Curves .....	97
2.5.9	Compare Traffic Routings .....	99
2.5.10	Airport Arrival/Departure Time .....	101
2.5.11	Compare Network .....	104
	Segment Load Comparison.....	106
2.5.12	Compare Airspace .....	107
2.5.13	Route Charge .....	108
2.5.14	Sector Workload (Macroscopic formula) .....	112
2.5.15	Sector Workload / Capacity (CAPAN like) .....	114
	Introduction to CAPAN like.....	114
	Description of Input / Output Data.....	114
	Running CAPAN like Module.....	116
2.5.16	Delay .....	122
2.5.17	Airspace PRU Complexity .....	124
2.5.18	Airspace Flight Time (minutes) .....	126
2.5.19	Scenario Economy .....	127
2.5.20	Airspace Value Merge (Complexity) .....	131
2.6	Add Menu.....	132
2.6.1	Add Airspace .....	133
2.6.2	Add Network .....	134
2.6.3	Add Terrain .....	136
2.6.4	Add Runway .....	137
2.6.5	Add Traffic Animation .....	139
2.6.6	Add Circle .....	141
2.6.7	Add LayerSet .....	143
2.7	Transform Menu.....	144
2.7.1	S06 to Text .....	144
2.7.2	S06 to Traffic Demand ( .exp ) .....	145
2.7.3	S06 to Network .....	146
2.7.4	So6 to List of Options ( .lox ) .....	147
2.7.5	So6 to PIN .....	148
2.7.6	So6 To Direct Route PIN .....	149
2.7.7	S06 to Scenario .....	150
2.7.8	Flow Demand to S06 .....	151
2.7.9	Matching Files .....	152
2.7.10	Lowering S06 FL .....	156
2.7.11	Airspace to TDV .....	157
2.7.12	Traffic Punching .....	158
2.7.13	Conflict Traffic Cut to Animation .....	160
2.7.14	Combine SO6, ARE or LOP .....	162
2.7.15	Sector_SLS to Center_SLS .....	164
2.7.16	Airspace 3D Envelop .....	165
2.7.17	Extract Traffic within Time Period .....	166
2.7.18	Network Extraction .....	168
2.7.19	Data file decompression .....	169
2.7.20	Network Coordinate Check .....	170
2.8	Pick Info Menu.....	171
2.9	Export Menu.....	172
2.9.1	TAAM .....	172
2.9.2	IPAS .....	173
2.9.3	RAMS .....	176



2.9.4 Selected Airspace to Excel .....	179
2.9.5 Network Points .....	180
<b>2.10 Import Menu.....</b>	<b>181</b>
2.10.1 RAMS .....	182
2.10.2 Escape to SAAM .....	183
<b>2.11 View Menu.....</b>	<b>186</b>
<b>2.12 Help Menu.....</b>	<b>187</b>
 <b>Chapter 3 SAAM Tool Bar</b>	 <b>190</b>
3.1 Tool Bar.....	190
 <b>Chapter 4 SAAM Techniques</b>	 <b>194</b>
4.1 Dynamic Command of SAAM.....	194
4.2 Using the Graphical Editors.....	195
4.2.1 Editing Airspace Volumes .....	196
4.2.2 Editing Airspace: Commands & Shortcuts summary .....	215
4.2.3 Editing Route Network .....	216
4.3 SAAM Graph module.....	226
4.3.1 Opening a graph .....	226
4.3.2 Graph properties .....	226
4.3.3 Scrolling through graphs .....	228
4.3.4 Series properties .....	229
4.3.5 Axis properties .....	230
4.3.6 Filter .....	231
4.3.7 Copy Paste a graph .....	232
4.3.8 Night view .....	234
4.3.9 Page layout .....	234
4.3.10 Printing graphs .....	238
4.4 Using the TDV Editor.....	239
4.5 Modifying the Assignment & Profile Parameters.....	241
4.6 Segment Management Rules.....	245
4.7 Assignment data validation process.....	252
4.8 Preparing a SAAM Presentation.....	256
4.9 Creating a SAAM animation.....	257
4.10 Adding graphical objects.....	262
4.11 Airspace Activation Editor.....	270
4.12 Configuration Editor.....	274
4.12.1 Main window : tree view .....	275
4.12.2 Loading/saving airspaces .....	278
4.12.3 Airblocks .....	280
4.12.4 Elementary sectors .....	280
4.12.5 Collapsed sectors .....	283
4.12.6 Centers (AUA) .....	285
4.12.7 Center Groups (AUAG) .....	289
4.12.8 Configurations .....	290
4.12.9 Opening schemes .....	293
4.12.10 Incoherent airspace dataset .....	295
4.12.11 Search .....	297
4.12.12 Log viewer .....	298
4.13 Rule Editor.....	299
4.13.1 Open a rule file .....	302

4.13.2 Edition .....	303
4.13.3 Search .....	305
4.13.4 FLC2 Syntax checking .....	307
4.13.5 Layer engine integration .....	308
<b>4.14 SIM User Manual.....</b>	<b>310</b>
4.14.1 The catalogue .....	312
4.14.2 Menus and functions .....	313
Preferences Dialog.....	317
4.14.3 Diagram components .....	320
Terminals.....	321
Data blocks .....	322
Processing blocks.....	324
Links .....	325
Loops .....	326
Sub-diagram.....	329
4.14.4 Building diagrams .....	330
Adding data blocks.....	330
Adding processing blocks .....	332
Linking elements.....	335
Adding loops.....	336
Editing diagram properties.....	338
Building output file names .....	339
4.14.5 Running diagrams .....	340
4.14.6 Tutorials .....	343
Tutorial 1: Scenario economy comparison.....	344
<b>4.15 SIM Extension Guidelines.....</b>	<b>346</b>
4.15.1 Data types extensions .....	347
4.15.2 Processes extension .....	349
SPI processes.....	349
Plug-in processes.....	353
How to choose the appropriate integration technique.....	357

## Chapter 5 Appendices 360

<b>5.1 Command Line.....</b>	<b>360</b>
<b>5.2 File Types.....</b>	<b>360</b>
5.2.1 aco - Airspace Colour file .....	360
5.2.2 Aircraft Performance Profile Map .....	362
5.2.3 Aircraft Performance Profile Data .....	362
5.2.4 are - Newmaxo ASCII Region file .....	364
5.2.5 arp - Airport .....	366
5.2.6 as4 .....	367
5.2.7 ase - ASCII Segment file .....	367
5.2.8 asx - ASCII Segment Extended file .....	368
5.2.9 cap - Capacity file .....	370
5.2.10 cdr - Conditional Routes file .....	371
5.2.11 cfg - Sector Configuration file .....	372
5.2.12 cls - Sector file from CAPAN-like .....	373
5.2.13 clt - Task file from CAPAN-like .....	374
5.2.14 conf - Conflict file .....	376
5.2.15 cos - Configuration Opening Scheme file .....	377
5.2.16 cost - Route Charge Cost file .....	378
5.2.17 crco - CRCO Intersection file .....	378
5.2.18 eep - Entry Exit Point .....	379
5.2.19 equi .....	380
5.2.20 exp2 .....	380
5.2.21 exp - Expand file .....	388
5.2.22 flc2 - Flight Level Constraints .....	388

5.2.23 fld - Flow Demand file .....	389
5.2.24 flf - Flight Info file .....	389
5.2.25 frp - Free Route Points file .....	390
5.2.26 gar - Gasel Airblock file .....	392
5.2.27 gr1 - Sector Load Input Chart Data file .....	393
5.2.28 gsl - Gasel Sector file .....	394
5.2.29 interieur .....	395
5.2.30 jse - Jet Stream Effect file .....	396
5.2.31 lay - Layer file .....	397
5.2.32 lop - List of Options file .....	397
5.2.33 lox - List of Options Extended file .....	398
5.2.34 lsr - List of SOP Runs file .....	399
5.2.35 mot - Military Opening Time file .....	400
5.2.36 Old SAAM are file .....	401
5.2.37 ord - Sector Order file .....	402
5.2.38 pin zin - Profile Input file .....	402
5.2.39 pl6 - PointLine6 file .....	404
5.2.40 pnl - Penalisation file .....	405
5.2.41 pre - Presentation file .....	405
5.2.42 qt5 - Quick T5 file .....	406
5.2.43 rlex - Route Length Extension Exception .....	406
5.2.44 seg - Segment file .....	407
5.2.45 sel - Sector Entry List .....	407
5.2.46 sls - Sector List file .....	408
5.2.47 slt - Graph filter file .....	408
5.2.48 soa - Area Definition file .....	409
5.2.49 so6 - Traffic file .....	409
5.2.50 sos - Sector Boolean Opening Scheme file .....	411
5.2.51 spc - Gasel Airspace file .....	412
5.2.52 ssl - SID and STAR list .....	413
5.2.53 t5 - Airspace Traffic Intersection file .....	413
5.2.54 tax - Turn Angle Exception file .....	414
5.2.55 tcr - T5 Cleaning Rule file .....	414
5.2.56 tdv - Three Dimensional Viewer file .....	415
5.2.57 ter - Terrain Data file .....	425
5.2.58 tim - Timeline file .....	426
5.2.59 tol - Bunching Capacity (Tolerance) file .....	427
5.2.60 traj - Trajectory file .....	428

## Index

429



# Chapter

---



1

## CHAPTER 1: Introduction to SAAM

**SAAM** is a copyright product that may only be installed on a PC or run from a network with the express authorisation of the EUROCONTROL .

**SAAM** (System for traffic Assignment & Analysis at Macroscopic level) is a European airspace design evaluation tool. It is used to model, analyse & visualise route network and airspace volume developments at local, regional and European-wide levels. Typically, it is used by airspace planners and designers to improve airspace and sector capacity.

SAAM was initially developed by EUROCONTROL's Airspace, Network Planning & Navigation Unit to help in the assessment of the future European Route Network. Subsequent development of its modelling capability has also provided users with a tool for presentations.

SAAM, which continues to incorporate new features and improvements, presently consists of four integral parts. These parts, which all deal with Airspace structure and traffic data, are:

- Modelling
- Optimisation
- Analysis
- Visualisation & Presentation
- Import & Export

### 1.1 What's new

This Topics gives new features or changes built in SAAM

➤ [V4.2.0 December 2012](#)

#### 1.1.1 SAAM V4.2.0 December 2012

### SAAM V4.2.0 Beta

#### General

- Better handling of unexpected crashes. From now a dialog appears on a crash which allows sending an error report to the development team or restarting the software. TRAC #1215.
- Fix spurious error messages from the SAAM generic launcher. TRAC #1198.
- Upgrade software building environment, some computation results may change a bit. TRAC #1220.

#### Layer Engine

- Add the possibility to edit layered airspace file with SAAM [airspace editor](#). In this context, a button "*Layer Apply*" will be available dialog to store the modifications in the layer dashboard. More information on the [Layered Airspace Edition](#) section. TRAC #1007.
- When the coordinate of a point is changed in one network, the point is automatically moved at the same position in other networks where it exist. TRAC #1233.

*Note: The display of the other networks in SAAM are not automatically refreshed. However, it is possible to refresh it manually with the new "Refresh Editors" action in the layer*

*dashboard toolbar.*

- Adds a new dialog to create a text report based on the content of the layerset. This dialog is accessible from the "Report" action in the toolbar. It is also possible to make a report of a single layer or folder by using the "Report" action in the context menu. TRAC #860, TRAC #1070.
- Add a new action to set the visibility state of an item in all defined configurations. This feature is available from the context menu of any item with the "Change item visibility in configurations..." option. TRAC #1221.
- Fix and improve the behaviour of the "Replace baseline" feature. TRAC #1224.
- Add a new action to copy the structure of the selected items. i.e. All the folders and layers without any modifications. TRAC #1219.
- The "horizontal consistency" popup now always appear on the screen where SAAM is opened and on top of every other windows. TRAC #1225.
- Remove the warning message which often appeared when a layerset was opened. Opening the same layerset twice now display another (more reliable) message which invite to open the data in "read-only" mode (no modification allowed). TRAC #1212.
- Fix problem when changing Airspace from Generic Editor, SAAM did not refresh content. TRAC #1217.

**Important note:** SAAM 4.2.0 introduces a change of layerset format to Airspace integration. Old layersets will be automatically converted the first time you save them, but after that, old SAAM versions won't be able to opened them anymore.

## Profile

Fix a bug which occurred in some rare cases and which lead to time inconsistencies in the produced traffic file. TRAC #1226

## Extract and merge traffic dialog

Fix a bug which occurred when there were spaces in the scenario directory path. TRAC #1216.

## Airspace / Traffic intersection

Allow multiple instance run simultaneously. TRAC #631.

## Assignment

Fix the detection of missing turn angles. (regression from V4.0.7) TRAC #1218.

## Airport Arrival/Departure Times

Redesign a part of the dialog to add [a new field](#) which allow setting the date to process within the input traffic file. TRAC #1245.

*Note for SIM users:* The version of the SIM process has been raised in order to add two new

input slots to handle the date. However, the old process (version 1.0) is still available under the "Deprecated" process folder for compatibility reasons with existing diagrams.

## Queries

- Re-integrate the [vertical profile viewer](#). TRAC #1238,
- skip '!'-prefixed technical points for good query results pertinence. TRAC #669.
- update code handling new so6 large flight IDs coming from DDR2 forecast according to new behaviour of compiler VS2008. TRAC#1237.
- info line displays a message when Query convert so6 from Unix to Windows end-of-line style.

## Menus

- Remove unavailable "Traffic Forecast" menu item. TRAC #402,
- Specify the North Compass : Geographic. TRAC #935.

### 1.1.2 SAAM V4.1.0 November 2012

## SAAM V4.1.0 Beta

### Layer Engine

- A new feature is available to import the content of a flat file into a layer. All the files editable with the generic editor plus ASE files are supported by this feature which is accessible from the context menu of a layer. TRAC #955 and #1195.
- A copy/cut/paste function has been added. It completely replaces the existing cut/paste feature for modifications. TRAC #1209.
- A new concept of *layerset configuration* has been introduced. A configuration is a definition of the visible state for each items (AIRACs, folders, layers or modifications) inside the layerset. It is possible to define multiple configurations and switch easily between them with the configuration menu located in the upper right corner of the dashboard. TRAC #1025.
- Displays next to each folders/layers a small summary of its content: the number of valid, hidden and invalid modifications. TRAC #1200.
- Improves the visibility of hidden items among a big layerset. Hidden items are now highlighted with a light grey background. TRAC #1136.
- Gives the possibility to add multiple line MOT items for a single sector in the generic editor. TRAC #1205.
- Fixes a crash which occurred when SAAM is refreshed and if the layer dashboard contained some unsaved changes. TRAC #1208.



- Fixes a crash which occurred when SAAM is closed and if the layer dashboard is opened. #1194.
- Fixes the context menu of a layer which displayed two "Rename" actions. TRAC #1197.

**Important note:** SAAM 4.1.0 introduces a change of layerset format to support configurations. Old layersets will be automatically converted the first time you save them, but after that, old SAAM versions won't be able to open them anymore.

## SIM

- Fix the management of optional input file slot. TRAC #1203,
- Some trajectory simulation diagrams (Assignment + Profile) have been added.

## 1.2 SAAM Requirements

### Operating System

SAAM is designed to run under Windows 2000, NT and XP and VISTA. Preferably XP

### User Knowledge

SAAM users should be familiar with the Windows environment and terminology used in ATC context.

### Hardware Requirements

SAAM requires a PC with the following minimum specification.

- A CPU with a speed of at least 1 GHz
- 512Mb of RAM (1Gb or more is recommended),
- A graphic card with 3D hardware graphic acceleration and a minimum of 16Mb of on-board memory.
- A hard disc drive with fast access and at least 8Gb of free space.
- A 19" monitor with a screen resolution of 1024x768 (A 21" monitor with 1280x1024 resolution recommended)
- A two buttons mouse (3 button type recommended).

### SAAM when using Windows Vista

You may encounter problems with SAAM when using Windows Vista. If that happens and some modules doesn't work you can activate the "Windows XP Compatibility Mode". "Windows XP Compatibility Mode" is a feature of Windows Vista that allows you to run programs written for earlier versions of Windows, for example, Windows XP.

To use Windows XP compatibility mode, right-click the executable (\*.exe) and choose Properties from the context menu that appears. A window will appear displaying the different properties of the application organized in tabs. Selecting the Compatibility tab allows for various, newer features of the operating system (Windows Vista) to be disabled when running

this program. Additionally, the Compatibility tab gives the option to run your application as it is intended for Windows XP (Service Pack 2) or for almost all other previous versions of Windows, as available from the drop-down menu. Click Apply then click OK to save the changes.

If the operating system compatibility mode is not enough, try running the "Program Compatibility Wizard" by clicking on Windows Vista Start Button, then Control Panel □ Programs □ Programs and Features □ Use an older program with this version of Windows.

## 1.3 Disclaimer

© Copyright 1992 - 2012 EUROCONTROL

The term "SAAM" or "SAAM software" includes all files provided in SAAM directory and sub-directories: all executables files, libraries, scripts and data files.

The term "Partners" refers to ANSPS members of EUROCONTROL or EUROCONTROL stakeholders or organisations with which EUROCONTROL has officially signed memorandum of cooperation.

The term "external company" refers to organisations or private companies that do not belong to partner's list and contracting for a limited period of time to perform studies or work under EUROCONTROL umbrella.

You should read carefully the following terms and conditions before using SAAM software. Downloading and installing this software indicates your acceptance of this licence agreement and warranty.

This is a free software. No ownership of intellectual property rights is transferred by virtue of this licence. You must not sell, copy, extract, trade, modify, enhance, convert , reverse compile, reverse engineer or translate SAAM.

### DISCLAIMER OF WARRANTY

SAAM AND THE ACCOMPANYING FILES ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OR QUALITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH SAAM MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE CAN BE OFFERED. ACCORDINGLY YOU ARE GIVEN NO WARRANTY THAT SAAM IS FREE OF VIRUSES OR OTHER HARMFUL COMPONENTS OR WILL BE ERROR FREE NOR THAT ANY ERRORS WILL BE CORRECTED.

NO WARRANTIES ARE IMPLIED AS TO THE COMPATIBLE FORMAT OF CURRENT OR FUTURE FLIGHT TRAJECTORIES AND THE ACCURACY OF THE RESULTS FROM THE USE OF SAAM.

YOU USE SAAM COMPLETELY AT YOUR OWN RISK AND YOU MUST MAKE YOUR OWN EVALUATION OF THE POTENTIAL USEFULNESS AND ACCURACY OF SAAM. EUROCONTROL EXCLUDES, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, ITS LIABILITY HOWSOEVER ARISING IN RESPECT OF ANY LOSS, DAMAGE, COSTS, CLAIMS OR EXPENSES HOWSOEVER ARISING SUSTAINED OR INCURRED BY YOU AS A RESULT OF YOUR USE, ATTEMPTED USE OR INSTALLATION OF SAAM AND THE ACCOMPANYING FILES.

IN PARTICULAR, EUROCONTROL SHALL NOT UNDER ANY CIRCUMSTANCES BE LIABLE TO YOU FOR ANY LOSS OF PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR SIMILAR DAMAGES (INCLUDING, WITHOUT

LIMITATION, DAMAGES FOR LOSS OF BUSINESS INFORMATION, DATA OR OTHER ECONOMIC LOSS) HOWSOEVER ARISING OUT OF YOUR USE, ATTEMPTED USE OR INSTALLATION OF SAAM.

THE PROVISION OF SAAM TO EXTERNAL COMPANIES IS SUBJECT TO TWO RESTRICTIONS:

- ONLY THE CURRENT VERSION OF SAAM IS PROVIDED. THE EXTERNAL COMPANY WILL NOT RECEIVE AUTOMATICALLY UPDATES OF SAAM. IF THE EXTERNAL COMPANY REQUESTS A NEWER VERSION, THIS REQUEST MIGHT NOT BE GRANTED.
- USAGE OF SAAM IS LIMITED IN TIME AND IN SCOPE FOR THE STUDY OR PROJECT FOR WHICH SAAM WAS PROVIDED.

## 1.4 User Support

### On-Line Reference Manual

The SAAM Reference Manual, this document, is available by selecting “Help/SAAM Reference Manual” on the SAAM main menu bar.

### SAAM Training Courses

The PowerPoint presentations used in SAAM training courses are available under “Help/SAAM Training courses” on the SAAM main menu bar. There are four different courses.

- Module 1 - Basic
- Module 2 – Scenario Processing
- Module 3 – Airspace Design
- Module 4 - Advanced Training

To attend a SAAM training course at Eurocontrol, go to SAAM web page.

<http://www.saam.aero/>

<http://www.eurocontrol.int/saam>

Under Training you can find the dates for planned courses and also a registration form.

### On-Line Video Tutorials

Video Clips showing various SAAM techniques are available by selecting “Help/SAAM video Training” on the SAAM main menu bar.

### SAAM Command line info

SAAM can be launched by a shell or a DOS command prompt. Information about this can be found in “Help/SAAM Command line info” on the SAAM main menu bar.

### Help Desk

Help Desk facilities are available at:

Eurocontrol

SAAM Support Team

Phone: (32) (0)2 729 35 34

E-mail: [saam@eurocontrol.int](mailto:saam@eurocontrol.int)

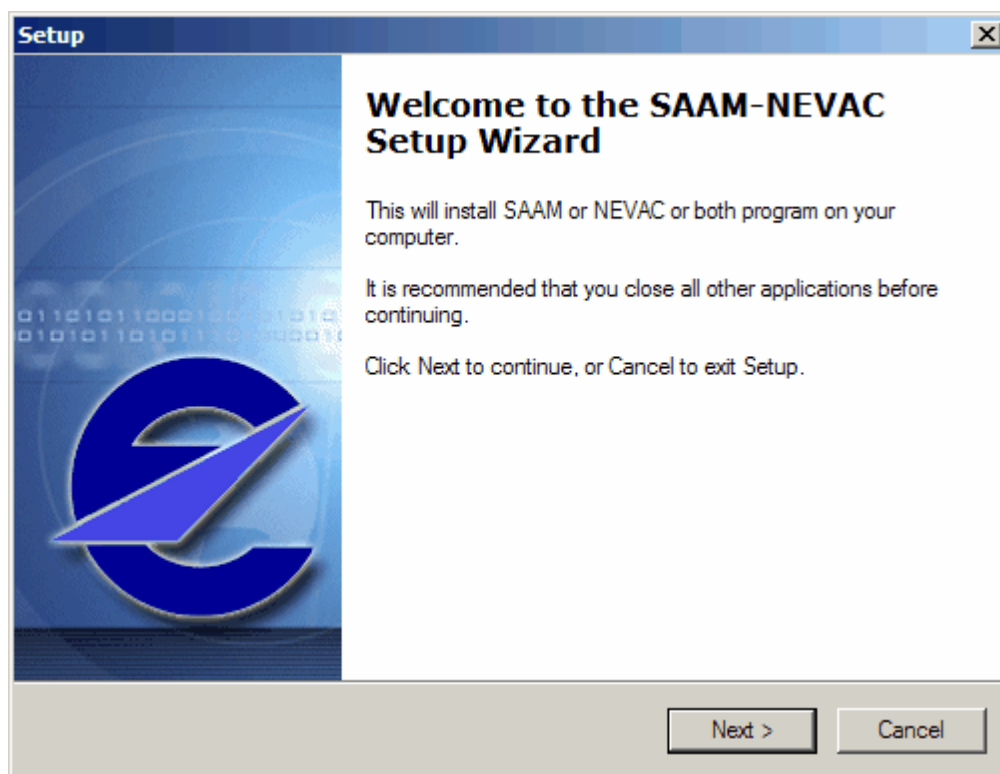
## 1.5 SAAM Installation

### Installing SAAM

The SAAM program can be down-loaded from Demand Data Repository (DDR) web application. You need to be granted access before you are able to down-load SAAM. Read more on DDR web page; <http://www.eurocontrol.int/ddr> .

Under the tab **Tools Download** you will find the latest SAAM release. Download and save somewhere on your local disk.

You have the possibility to install SAAM and NEVAC at the same time. The two softwares are now distributed under the same setup to facilitate future communications between the two platforms. It is not mandatory to install both of them, you can choose one or the other or both.



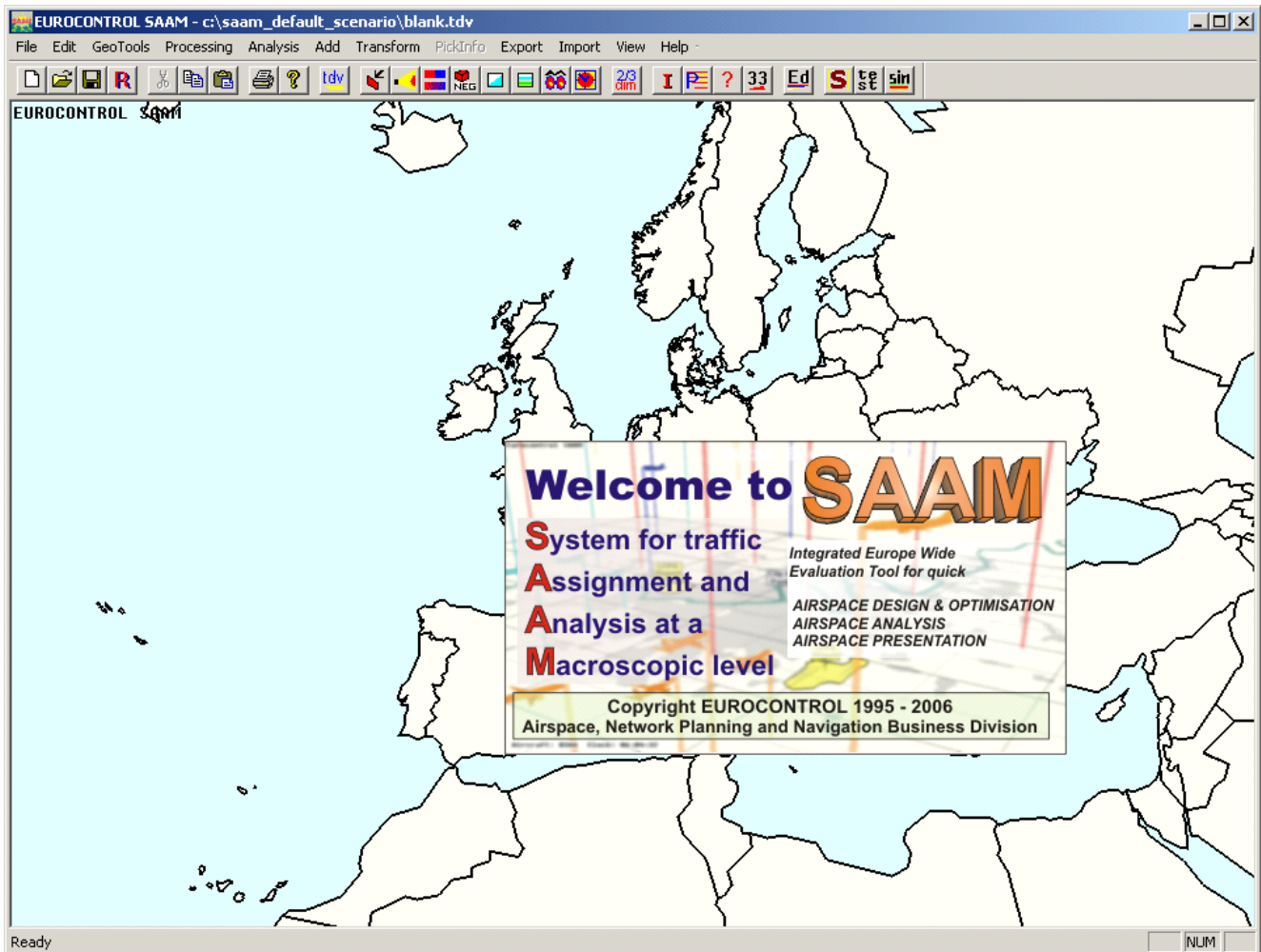
The SAAM program is completely self-contained, and relies entirely on programmes that are delivered in the SAAM directory, and do not touch any features in the system on your computer.

### Desktop Icon

A SAAM desktop icon is placed on the desktop. This is a short-cut to the file SAAM.exe found in SAAM directory.

## Starting SAAM for the first time

Double-clicking on the SAAM start-up icon (or on SAAM.exe) will produce a start-up screen showing a default scenario called blank.tdv with a map of Europe. The blank.tdv file is located in a default SAAM working directory called SAAM\_default\_scenario.



Start-up Screen

## 1.6 SAAM Data files

### Introduction

SAAM is using data files from different sources as input. This chapter provides a description of the most important ones.

A description of file format for files used by SAAM can be found in Appendices/[File Types](#)

### Traffic data

Traffic data files has the extension \*.so6. The traffic data normally comes from CFMU. There are two sets of traffic data available, model 1 and model 3.

- Model 1 is pure flight plan data, as filed by the airlines and corrected by CFMU.
- Model 3 is flight plan data compared and changed according to radar data, but the flight

route is still described from navaid to navaid.

## Airspace data

SAAM is able to read two different sets of Airspace data, ARE / SLS files and Gasel format. Both formats are translations of CFMU environment data.

ARE / SLS files, is the most common file format used.

- Are file, \*.are, contains the coordinates, and the
- SLS file, \*.sls, contains the lower and upper FL of volumes and association of volumes.

The Gasel format have several files to describe an airspace.

- Airblock file, \*.gar – 2D definition of airblocks
- Sector file, \*.gsl – How elementary sectors are built from airblocks
- Airspace file, \*.spc – How elementary sectors can be collapsed, can also contains description other bigger airspace (like ACC ...)
- Configuration file, \*.cfg – Different sector configurations used by ACCs
- Configuration Opening Scheme, \*.cos – The start and end time when different sector configurations are used.

## Network data

The network data mostly used in SAAM is called VST network. Where VST stands for Very Short Term. It is a network containing all changes to the European route network agreed by Route Network Development Sub-Group (RNDSG)

## Conversion DecimalMinutes to Lat/Long and vice versa

In all SAAM files coordinates are described in DecimalMinutes, It is easy to convert DecimalMinutes to Lat/Long and vice versa using Excel.

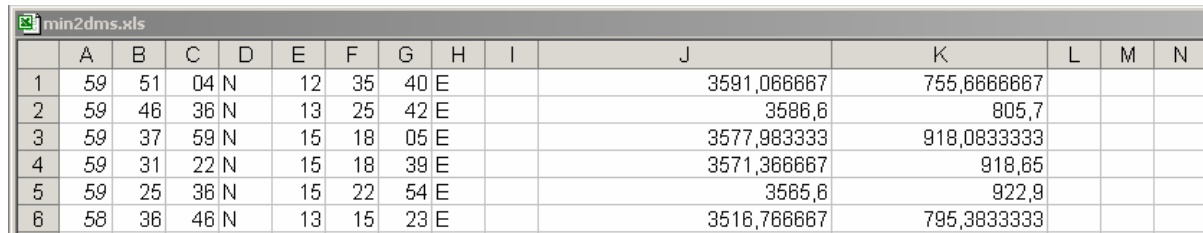
min2dms.xls															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	3.063,48	249,85		51	N	03	29		04	E	09	50		51 N 3 29	4 E 9 50
2	3.398,15	-1.024,29		56	N	38	09		17	W	04	17		56 N 38 9	17 W 4 17
3	-681,25	1.010,47		11	S	21	15		16	E	50	28		11 S 21 15	16 E 50 28
4	2.956,16	13.405,13		49	N	16	09		223	E	25	07		49 N 16 9	223 E 25 7
5	-1.476,28	-5.443,79		24	S	36	16		90	W	43	47		24 S 36 16	90 W 43 47
6	-167,98	6.092,54		02	S	47	58		101	E	32	32		2 S 47 58	101 E 32 32

### DecimalMinutes to Lat/Long

Insert funktion (fx);

- Column D      fx = INT(ABS(A1)/60)
- Column E      fx = IF(A1<0;"S";"N")
- Column F      fx = INT(MOD(ABS(A1);60))
- Column G      fx = INT((ABS(A1)-INT(ABS(A1))))\*60)
- Column I      fx = INT(ABS(B1)/60)
- Column J      fx = IF(B1<0;"W";"E")
- Column K      fx = INT(MOD(ABS(B1);60))
- Column L      fx = INT((ABS(B1)-INT(ABS(B1))))\*60)
- Column N      fx = D1&" "&E1&" "&F1&" "&G1&

- Column O       $fx = I1 \& " "& J1 \& " "& K1 \& " "& L1 \&$



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	59	51	04 N		12	35	40 E			3591,066667	755,6666667			
2	59	46	36 N		13	25	42 E			3586,6	805,7			
3	59	37	59 N		15	18	05 E			3577,983333	918,0833333			
4	59	31	22 N		15	18	39 E			3571,366667	918,65			
5	59	25	36 N		15	22	54 E			3565,6	922,9			
6	58	36	46 N		13	15	23 E			3516,766667	795,3833333			

***Lat/Long to DecimalMinutes***

Insert funktion (fx);

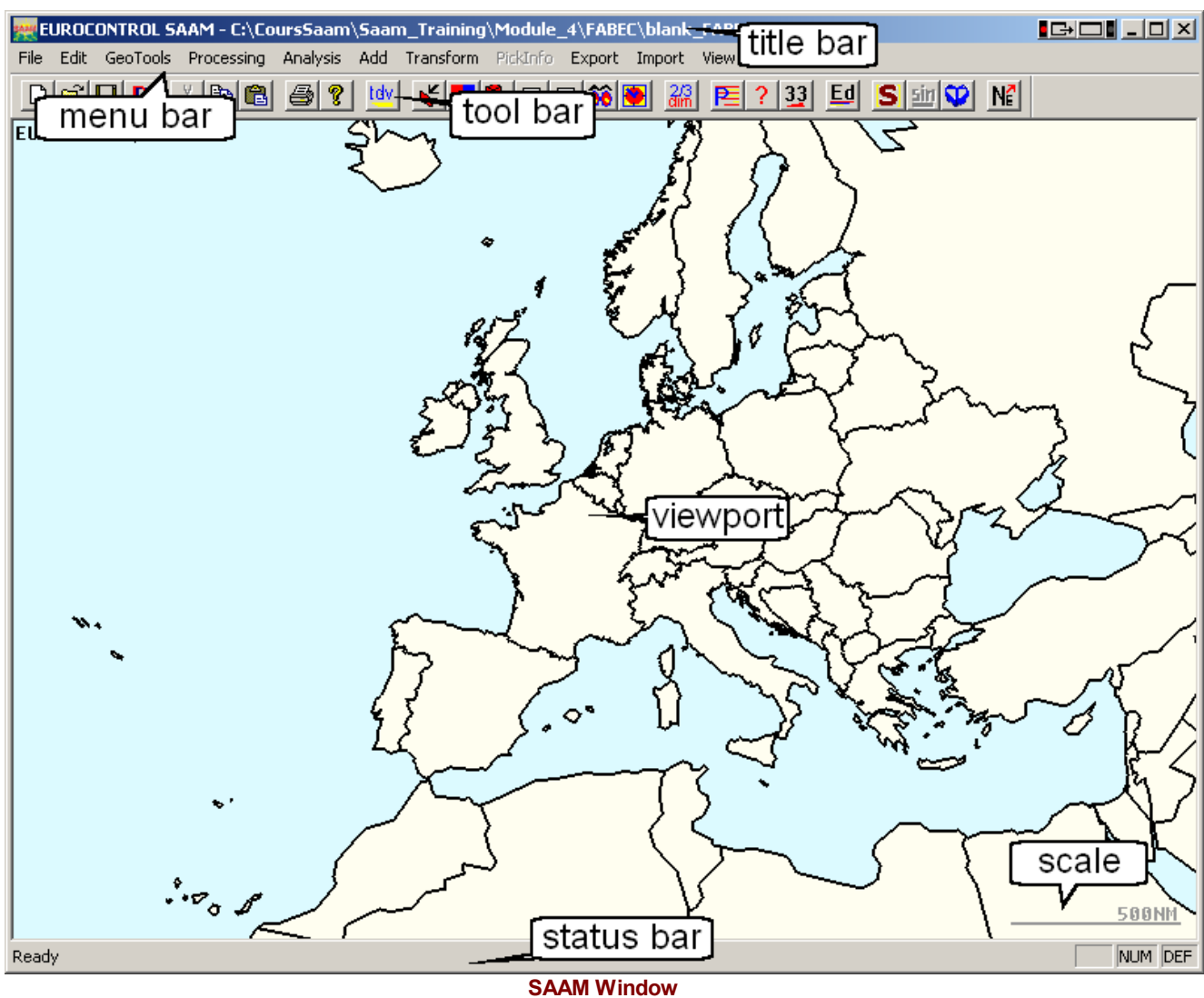
- Column J       $fx = (A3 * 60) + B3 + (C3 / 60)$
- Column K       $fx = (E3 * 60) + F3 + (G3 / 60)$

## 1.7 The SAAM Window

### Introduction

The SAAM window contains

- title bar
- menu bar
- tool bar
- viewport
- status bar
- scale



## Title Bar

The title bar will behave like any other window application to show when it's active and will contain the path of the current scenario TDV file.

## Menu Bar

The menu bar provides the user access to various SAAM functions and facilities (see [Menu Bar](#))

## Tool Bar

The tool bar contains icons, which allow the user a single click access various SAAM tools (see [Tool Bar](#))



## SAAM Viewport

The SAAM viewport occupies most of the SAAM window and is used to display the 2D/3D views of maps and traffic.

## Scale

The scale is expressed in NM and is only visible in top view (menu View/Set Top View). It doesn't matter if your map is displayed in 2D or 3D. It will be printed on your map. The scale is auto re-sized from 0,1 NM to 5000NM.

### 1.7.1 Manipulating the Picture

The mouse is used to change the view of the picture on the screen. The technique in using the mouse to change the view, is to click and hold the appropriate button and then move the mouse to re-scale or re-position the picture.

#### Zoom In / Zoom Out

Click and hold the left mouse button. Move the mouse horizontally.

Note: Zoom speed depends on the distance to the ground reference level. The closer you get to the ground reference level, the slower it will be to zoom in.

#### Rotate and Tilt the Picture

Click and hold the right mouse button. Move the mouse horizontally to rotate the picture. To tilt the picture move the mouse up and down.

#### Change the Centre of the Picture

Click and hold both mouse buttons (or middle button) to change the centre of the picture.

#### Default Values

By pressing the '**Home**' key the view will change to the default values, i.e. the start-up view.

#### Move the picture vertically

To rotate and tilt an object located up in the sky, it's necessary to change the ground reference level.

The left button has an additional function, which is activated/deactivated by briefly pressing '**Ctrl+1**' and moving the mouse up and down. The cursor will change into an arrow indicating that the left button may now be used to move the picture vertically. What you actually do is moving the ground reference level from initial default level to up in the sky, and this change will affect all the following manipulations of the picture, so be careful of how you use this feature. It will also have an effect on the zoom-in speed. To revert to the normal mode, briefly press '**Ctrl+1**' once and move the mouse.

## Viewing a Continuous Picture Movement

The display will move continuously if '9' on the numerical pad is pressed. Moving the mouse in any direction (rotation or zoom in/out) will initiate a continuous movement of the display.

To brake slowly use '1' of the numerical pad. To stop and disconnect the option use '0'

On laptops, you must use the Fn...Numlock (the blue digits) to access this function.

## Viewing full screen

Press 'F5' key to view SAAM in full screen. This can be useful when doing presentations. To change back press 'F5' again.

### 1.7.2 SAAM Working Directory

The default working directory, where SAAM normally loads and saves files, will automatically be located where the user chooses to open or create their own TDV file(s).

# Chapter

---



2

## CHAPTER 2: SAAM Menu Bar

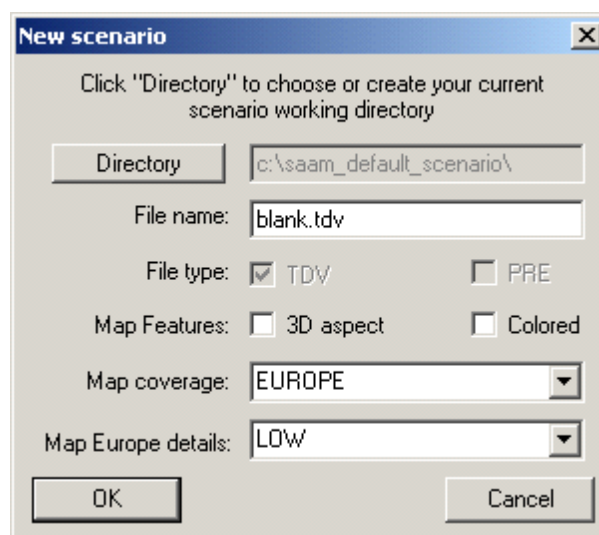
The menu bar provides the user access to various SAAM functions and facilities. This chapter provides a detailed description of each item on the menu bar.

### 2.1 File Menu

The file menu contains the following items;

- New
- Open
- Open Graph
- Layer with offers sub choices:
  1. Start a local project
  2. Projects Validation
- Save
- Save As
- Refresh
- Set as Default Scenario
- Print
- Print Preview
- Page Setup
- Recently used file list
- Quit

#### New



*New opens the New scenario window*

#### Parameters

1. **Directory field** – the working directory for the current SAAM session
2. **File name field** – the scenario file name.

3. **File type radio button** – TDV file is the only implemented option
4. **Map Features radio buttons** – **3D aspect**, when checked gives an impression of coastline by lifting the countries above the sea level. On slower machines it is recommended not to use 3D aspect.. **Colored**, will give all the countries an individual colour.
5. **Map coverage drop-down menu** – the default map that will be used, Europe or World. On slower machines it is recommended to use the Europe map, as it is less demanding for the computer.
  - a. **EUROPE** contains all of the European Countries
  - b. **WORLD** comprises the whole Northern Hemisphere
6. **Map Europe details drop-down menu** – You can choose between **LOW** or **HIGH** details when selecting Europe as map. This is useful when a high detailed map can slow down animations.

## Open

Used to open a TDV file (\*.tdv), Presentation file (\*.pre) or a Scenario file (\*.sce). To open a Presentation or Scenario file you have to change “Files of type” to the correct one.

## Open Graph

Opens a graph that has been created using the [SAAM graph](#) generator (\*.gr1) ( see [Airspace Load](#) )

## Layer

Offer 2 sub categories:

- Start a local project. It will open the Layer Dashboard on the shared layer directory in read only mode. From here, users have the choice to start a new project (right click on a project or an AIRAC cycle). An artificial baseline will be compiled and sent to a local directory (in the current working directory) called "Local LayerSets", where a new local project will be created. At this time the Layer dashboard will display a second project: your local read & write Layer project. User can start to work. When finished, user must integrate his projects back into the Shared Layer directory, by right clicking on his project name in the Shared Layer directory and select "integrate". Next step will be the validation with all other projects, only available for user entitled to proceed to project validation (see below).

- Projects validation: it will open the Layer Dashboard on the Shared directory in read & write mode. So that it is possible to track the "vertical" consistency (between projects) and either proceed to correction or advice colleagues responsible for local projects, so that they themselves proceed to corrections.

Caution: It is not recommended to use Layer while current TDV is running an animation, or, if SAAM freezes, try to iconize it.

## Save

Saves the current TDV file

## Save As

Allows the user to save the current TDV file using a different name or in a different directory. All data files attached to the TDV file will also be saved (animation, terrain data, curve data ...) in the same target directory. The location references inside the 'save as' TDV file will be changed to the new location of the attached data files, so that it can be re-loaded without problems.

## Refresh

Are used to refresh the screen. Modifications that are not saved will be lost. This is equivalent to reopen the file. Warning: Some dialog boxes will close and all the settings will be lost.

## Set as Default Scenario

Allows the user to select an alternative TDV file as the default scenario. The procedure is to open the alternative TDV and then click on **Set as default scenario**. This file will then be loaded as the default scenario when starting SAAM.

## Print

Prints the active screen.

After clicking 'Print' in the print dialog box a new dialog box will open. There you can choose the print quality and number of pages your image should be printed at. You can also select different file formats (BMP, PNG and JPG) which is only relevant if you have chosen 'Print to file'.

If '4 pages' is selected in 'Number of page(s)', 1/4 of the image will be printed on one page.

If in 'Print quality/page' you select 'ROUGHx1', there will be the same number of pixels on the printout as there are on your screen, and if 'NORMALx3' is selected the number of pixels on the printout is multiplied by three for the width and three for the height.

## Print Preview

Not implemented.

## Page Setup

Not implemented.

## Recently used file list

Lists the files you've most recently opened in this program. To quickly reopen one of these files, click it.

## Quit

Closes the SAAM program.

## 2.2 Edit Menu

The Edit menu is in development, so for the moment these commands are only available for some actions in [Airspace Editor](#).

- Undo
- Cut
- Copy
- Paste

### Undo

Reverses the last command or deletes the last entry you typed.

### Cut

Removes the selection from the active document and places it on the Clipboard.

### Copy

Copies the selection to the Clipboard.

### Paste

Inserts the contents of the Clipboard at the insertion point, and replaces any selection.

## 2.3 Geo Tools Menu

The Geo (*Geographical*) Tools menu contains the following menu items;

- Where-am-I
- Length/Azimuth
- North Compass

### Where-am-I

By clicking on 'Where-am-I' and then clicking on the map, you get the position of the mouse pointer transformed into geographical coordinates expressed as Lat/Long and Minutes. To switch-off the function you click on 'Where-am-I'. This feature can be used while editing airspace or network.

### Length/Azimuth

By clicking on 'Length/Azimuth' and then click two points on the map, you get the distance between them in Nautical miles and Kilometres. The length is calculated along a great circle. You also get the Azimuth in degrees. To switch-off the function you click on 'Length/Azimuth' again. This feature can be used while editing airspace or network.

## North Compass

When clicking on 'North Compass' and then move the mouse pointer over the map, you'll get a line indicating Compass-direction North. To switch-off the function you just click on 'North Compass'. This feature can be used while users operates on all SAAM functions.

## 2.4 Processing Menu

### 2.4.1 Assignment

#### Accessed by

[Menu Bar / Processing / Assignment](#)

#### Purpose / Description

Find routes on a given network for a given traffic demand. This process is called Assignment, by default the routes found are the shortest ones.

A network is made up of several connected segments. A route consists of one or more segments. The Route Length (RL) is the sum of the Segment Length (SL) for all segments that the route contains. The assignment process always automatically chooses the routes with the shortest length.

$$RL = SL1 + SL2 + SL3.....$$

Traffic demand is found in a traffic demand file (\*.exp), which contains number of flights between each city-pair.

Assignment process generates a ZIN file for 3D profiling processes, which is mandatory for a complete 4D aircraft trajectory processing.

#### Input

- A traffic demand file (\*.exp or \*.exp2). It is recommended to use \*.exp2 traffic demand file, as it is the way to convert, if necessary, trajectories into ALLFT files (used by NEVAC & other simulators). How to get \*.exp2 traffic demand ? Through the DDR, it is possible to either download past traffic demand or to generate future traffic demand. See [Import Menu](#) and [DDR access](#). Remark : IFR flights operated with an helicopter may not be of interest. See [excluding helicopters](#) to put them apart of you traffic demand.
- A description of the route network which is a set of structural network description (\*.ase file, or \*.frp for automatic Free route segments generation) with associated conditions of use (turn angle exceptions (\*.tax file), segment management rules (\*.awk file), SID/STAR network connection points (\*.sid, \*.star files), level and time conditions (\*.are, \*.sls, \*.mot files), penalisations (\*.pnl file)),
- Airport coordinates (e.g. airports2008.arp file),
- ZOD definitions (\*.awk file),
- Area definitions (\*.soa file),
- Wind file (\*.awk file), to simulate the effects of Atlantic Wind,
- Route Length extension exception file (\*.rlex file) allow extra route length extension for some city pairs,



## Output

- a loaded network file (\*.ase). A network where number of flights (load) on each route segment is known. This output file is not generated when Route Option Generation is run at the same time.
- an intermediate file (\*.zin) that will be used as an input for [profile calculation](#)
- if some flights are not assigned or badly assigned (which mean flights route length extension overpass the given limit) the file \*\_missing.exp is generated. It will contains all of these flights and can be used as an input for re-injecting flights using [matching](#) process (see example 2).
- a log file (\*.log) is output as well, warning for inconsistencies in the input files, overtaking of allowed route extensions, overtaking of maximum number of options, impossible airport/network connection...
- optional additional output files : a 2D so6 file, and detailed output files for validation process (see [below](#)).
- if [Route Option Generation](#) is requested: a list of option file (\*.lox) if generated.

## General Procedure

1. Prepare and group the input files into a working directory
2. Load the input files into the appropriate fields
3. Specify the scenario name
4. Edit input files and/or modify parameters (*optional*)
5. Launch the Assignment process
6. Check the assignment.log file

**Assignment**

**Inputs**

Traffic demand(s) (*exp2*) (1/1) 151158871.exp2  

RFL selection for profile estimation FromTrafficDemand

**Network**

☐ Close all temporary CDRs (2, 3, 1-2, 1-3)

**Airports management**

Airports coordinates (*arp*) pn\_a/VST\_1205\_Airports.arp  

☒ Area definitions (*soa*) nition\_RAD\_Appendix\_2.soa  

**Turn angle**

☒ Limit the turn angle at 90°

**Wind rules**

☒ Wind rules (*awk*) etstream\_Atlantic\_Wind.awk  

**Extension validity**

Extension of 150 NM or 70% for regular flights.

Extension of 100 NM for circular flights.

☒ Exception file (*rlx*) option\_Extension\_Validity.rlx  

**Outputs**

Scenario name eli2a2c

**Generated files**

Loaded network (*ase*) (1/1) ./151158871\_VST\_1205\_eli2a2c\_ld.ase

☐ 2D traffic (*so6*) (No output)

Profile input file (*zin*) (1/1) ./151158871\_VST\_1205\_eli2a2c.zin

Options (*box*) (No output)

Missing flights (*exp2*) (1/1) ./151158871\_VST\_1205\_eli2a2c\_missing.exp2

LOG (1/1) ./151158871\_VST\_1205\_eli2a2c.log

☒ Open the log automatically, if there are errors.

☐ Additional files for data validation

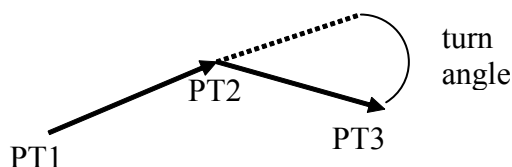
**Assignment Dialog Box**

## Parameters:

1. **Input fields** - All file names are selected using the Browse button
2. **Edit buttons** – Editing \*.exp, and especially \*.exp2, files is normally not recommended. Network files (\*.ase) may be edited via the [Network Editor](#)
3. **RFL selection for Flight Profile estimation radiobutton** –

Since non VST segments may be restricted in level or time, flight profile during assignment has to be evaluated somehow. Normally, flight profile is precisely computed during Profile processing (see [Profile](#)), taking into account Flight Level constraints. This is not possible during 2D assignment. As a consequence, the profile estimation used during assignment is constraint-free. That is : Flight profile falls into 3 parts : continuous climb, cruise, and continuous descent. As a consequence, after proper profile computation, non VST flown segments may not fully comply with the level and time restrictions. But in order to be as close as possible, RFL has to be selected in accordance with what will be chosen during Profile processing performed on Assignment output data.

4. **Network parameters button** – Opens the dialog box, where input files and [parameters](#) are selected for using VST/DCT/Night/Free route networks.
5. **Route option generator parameters button** – Opens a dialog box, where input files and parameters are selected for running the [Route Option Generator](#).
6. **CDRs management checkbox** – When **Close all temporary CDRs (2/3/1-2,1-3...)** is checked, all CDRs specified as CDR3, CDR2, CDR1+2 and CDR1+3 are closed and can not be used in the assignment process.
7. **Airport Coordinates field** – A text file that contains all airport co-ordinates.
8. **Airport To Zone checkbox and field** - In order to facilitate expression of rules, several airports can be grouped inside a ZOD, which bears the name of the most important airport in the zone. Grouping parameters can be found in the AirportToZone description file (see [Using Airports or Zones](#)). This also apply for SID/STAR assignment.
9. **Area Definitions checkbox and field** - In order to facilitate expression of rules, several airports can be grouped in areas. These areas are for example defined in RAD appendix 2.
10. **Turn Angle checkbox and field** – used to force the path finder to respect a maximum turn angle between two consecutive route segments:

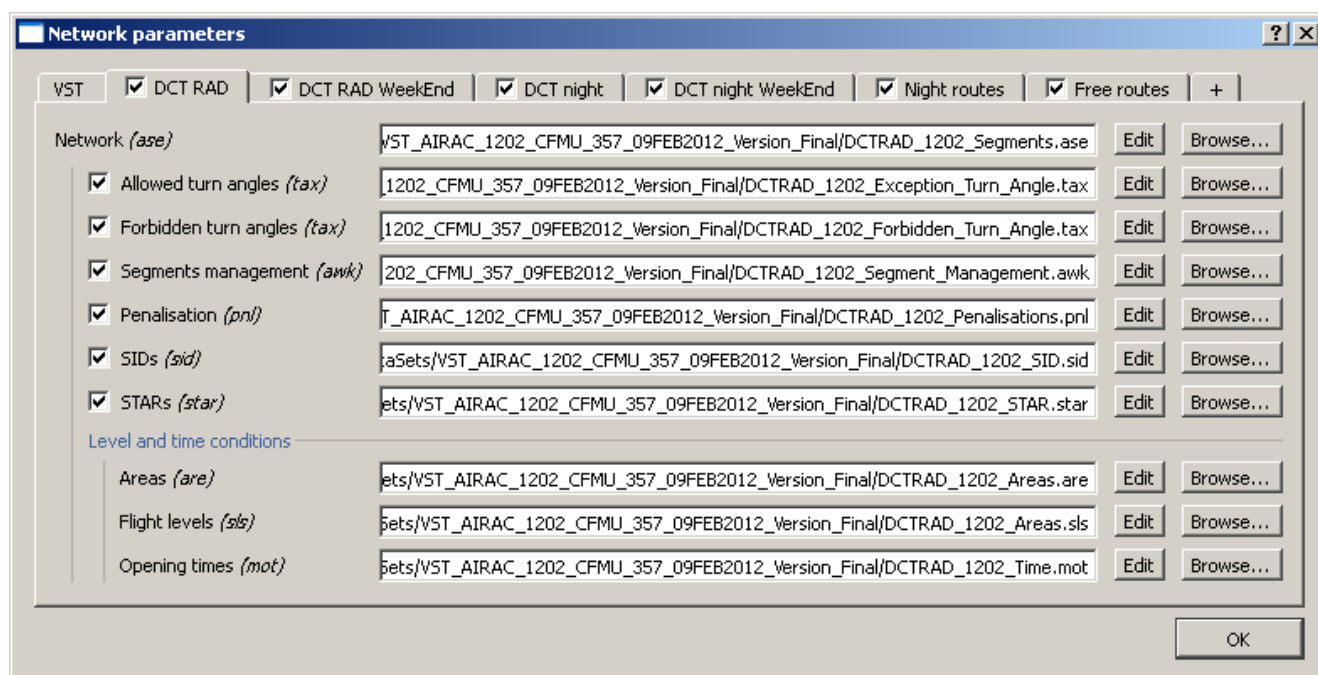


11. **Wind rules checkbox and field** – To simulate how Atlantic jet stream impact on chosen flight route. A \*.awk file following the same syntax as the one for Segment Management rules (see [Modifying the Assignment Parameters](#)).
12. **Extension Validity frame** – If the extension of the suggested route compared with the direct route is longer, in percentage or in Nautical miles than the given values, then the flight will not be assigned.
  - In percentage field**
  - In NM field**
  - for circular flights in NM field** – default route length extension allowed when ADEP and ADES are identical
  - Exception file checkbox and field** – It is possible to give a specific extension validity for some city-pairs. See [\\*.rlex file format description](#).
13. **Output fields** - By default the names of the Expand file and the Network file are copied to the Output boxes.
14. **Scenario name field** - If you give a name for the **Scenario**, that name will be added to the Output file names.
15. **Loaded ASE file field** - Displays the name of this output file. This file is not generated when running the [Route Option Generator](#).
16. **2D SO6 files checkbox** – When ticked, a \*\_2d.so6 2D trajectory file which displays the

trajectories on the requested flight level and corresponding to the output .zin file, will be generated.

17. **Open the log automatically checkbox** – When ticked, the log file will be automatically opened when the assignment process has detected some error in the data.
18. **Additional files for data validation process checkbox** – This checkbox is enabled provided that option generation is not used. If ticked, a set of additional files will be generated containing information about what flights were too long, for what possible reasons, what are the other missing flights, ... See [Assignment data validation process](#) for more information.
19. **LOX file field** - When input files are selected for the [Route Option Generator](#) the name of the output file/s will be displayed here.
20. **Run button** - Launch the Assignment process
21. **Stop processing button** – Stop the flight processing during the current expand files being processed
22. **Stop batch button** – Stop the processing queue when several expand files are to be processed
23. **Close button** – Close the assignment dialog box

### 2.4.1.1 Network Parameters



**Network Parameters Dialog Box**

## Parameters

All of these parameter files are text files which may be edited by the user. The chapter [Modifying the Assignment Parameters](#) gives an in-depth analysis of the various assignment parameters. The window is a property sheet where each Network dataset can be edited and selected independently. Tabs are used to navigate from one Network (VST/DCT/Night/...) to the other one. A Network tab labelled in blue colour means that this network is not used.

Tab with a +, gives you the possibility to add new network or associated data. For these extra tabs, network is not mandatory. Click on the +, and name your Network dataset. The name

has to be unique and there is a limitation for five new tabs. Extra tabs can be removed using the "Remove" button. Both explicit network and free route network can be added this way.

A Network dataset has to be consistent : network items (segments, points) referred to in Turn angle exceptions file, penalisation file, segment management file, SID/STAR file have to exist, at least partially, in the network definition file. Consistency checks are not done for files specified in extra tabs.

Non VST networks are published route segments (or Free route generating points) that can be activated for a period of time and a level band within an area. These networks have to be connected in some way to the mandatory VST network but can define and connect to additional points. These networks are described separately and will extend the VST network which is not level or time limited.

1. **Network checkbox and field** – used to select a network. Note that the use of VST network is mandatory. Note also that the Network file is replaced with a Free Route Points file in the Free Route tab. This Free Route Points file is used to generate an additional network on the fly (see [Free route network generation](#)).
2. **Allowed Turn Angle Exceptions checkbox and field** – This file gives you a tool to allow certain route segments despite of the turn angle limitation, by listing specific authorized sequences of 3 points. Each line in this file ([\\*.tax](#)) has the following format : <PT1> <PT2> <PT3>
3. **Forbidden Turn Angle Exceptions checkbox and field** – This file gives you a tool to forbid certain route segments in addition to the default turn angle limitation, by listing specific forbidden sequences of 3 points. Each line in this file ([\\*.tax](#)) has the following format : <PT1> <PT2> <PT3>
4. **Segment Management checkboxes and fields** – The 2D Assignment Rule file (\*.awk), contains rules which bypass the shortest path function by restricting segments to specific city-pairs or by forbidding other segments to other specific city-pairs. Users are warned not to modify this file unless they are very familiar with route networks and assignment processes.
5. **Penalisations checkboxes and fields** – During the assignment a Penalisation file (\*.pnl) is a way to change route lengths. This file contains for each specified segment (S) a value, expressed in NM, called penalty (Ps). This penalty will be added to the true length of the segment (SL).

**penalised SL = SL + Ps**

The assignment process automatically chooses the route with the shortest length. Since the Route Length (RL) is the sum of the Segment Length (SL) for all segments that the route contains, you can change a route length by using penalties

**true RL = SL1 + SL2 + SL3 + SL4....**

**penalised RL = SL1 + (SL2 + P2) + SL3 + (SL4 + P4)....**

A positive penalty (Ps) will make the segment less 'attractive', because it will be longer.

It's also possible to have a negative value for the penalty (Ps). This value must never make the penalised segment length (SL) negative. A negative penalty (Ps) will make the segment more 'attractive', because it will be shorter. Negative penalties may cause the found route not to be optimal. In this case, a warning is issued in the log file.

6. **SID/STAR checkboxes and fields** - These files consists of specified arrival and departure points for an airport (see [SID and STAR Points](#))
7. **Areas/Flight levels/Opening time fields** - These files are used to set the allowed flight levels and times for the related network segments :
  - a. Enable checkbox – To enable/disable level/time restriction. Note that level/time restriction is never used with VST network and always with DCT RAD/DCT Night/Night Routes/Free routes networks. That's why the checkbox is greyed out in these cases.
  - b. Areas (.are) field – Level and time restrictions for segments are defined for a set of segments at once defined by the 2D area they cross.
  - c. Flight levels (.sls) field – Used to define the allowed level band for segments crossing the

associated 2D area.

- d. Airspace Activation time (.mot) field – Used to define the allowed time window for segments crossing the associated 2D area. Beware : version 1 or 3 of mot data is used since levels are defined in sls file. And unlike [Route Choice Based On Military activity](#), the time windows defined here specify the availability of segments for civil traffic, not the time when they are closed. MOT files are editable with the [Airspace Activation Editor](#) (by default) or with a text editor.

## Remarks

1. one and only one level band/time window specification is allowed per area. So, there must be a perfect match between airspaces mentioned in Areas, Flight levels and Opening time files,
2. if a non VST segment crosses several areas with different level/time restrictions, then the most restrictive level/time conditions are taken into account,
3. if a non VST segment doesn't cross any area defined for this Network dataset, the segment is closed without notice,
4. only one level band/time window is associated to a non VST segment during computation. In case of contradictory restrictions coming from different areas or different Network datasets, the segment is closed (a warning is issued),
5. if the loaded network is to be output (no route option generation), then any segment described in the different Networks will appear in the loaded network output file,
6. time windows in the mot file are defined cyclically for every day. First time is the beginning of the opening time window whereas second time is the end, no matter where midnight is located in between. Examples for night opening times (see [mot file format](#)): 2200 0400 stands for "from 10:00 pm to 04:00 am", 0100 0500 stands for "from 01:00 am to 05:00 am", 2000 2400 stands for "from 08:00 pm to midnight",
7. by following the nomenclature of data files used in DDR, it is possible to set all the parameter files for a network at once, just by browsing for the corresponding Network file (or Free route point file). What's more, it's possible to set all the parameter files for the whole dataset at once just by browsing for the VST network file.
8. DCTRAD and DCTNight network tabs exist in two versions : the week days version (from Monday to Friday) and the weekend days version to take into account weekend routes. The use of week days or weekend days data for DCTRAD and DCTNight networks is automatic, based on the timing of each flight. Segment management files, turn angle exceptions files and SID/STAR files are not expected to be used on the weekend tabs.
9. When a file is invalid or missing, its field and its tab will appear in red. When clicking on the OK button, a warning message will be display to remind you all the invalid network dataset. But you can choose to ignore this warning by clicking on "continue".

## 2.4.1.2 Free route network generation

### Purpose/Description

Will extend the normal network with Free route auto-generated segments associated to a Free Route Area. These segments may be restricted in flight level and in time depending on the area. The assignment will be the shortest route from origin to destination taking into account all possible Free Routes areas.

These segments are generated thanks to a Free route points (\*.frp) file which contains a list of Points used to generate segments between them according to the following process : referring to the file format description (see [Frp format](#)), every Free route segment from one entry point (E or D) defined for an area to an exit point (X or A) defined for the same area is generated. Intermediate points (I) are used to take into account possible Military areas inside Free route areas : these points are located around Military areas and will be used during Route Options Generation (see [Route Option Generator](#) ) to provide a lot of possible direct Route Options avoiding or not Military Areas, in addition to existing network (see [Route Choice Based On Military](#) for further info on this subject).

### Remarks

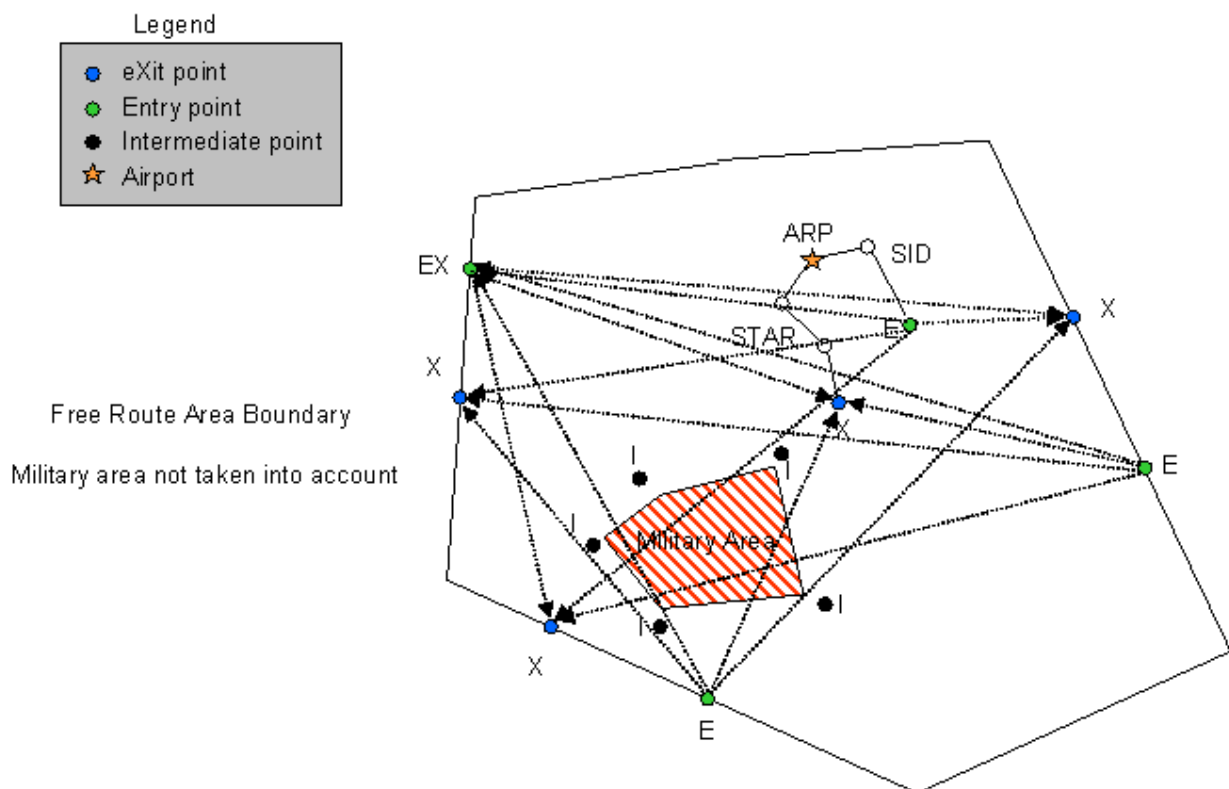
1. any generated segment which is already existing in declared networks (VST, DCT, Night, ...) won't be used under Free routing conditions, as if it was not generated,
2. only one definition is allowed per E/X/I/EX Free route point and per Area. The same Free route point may be defined differently a second time, but then for a different area (e.g. X for Area 1 and E for Area 2). As far as A/D/AD points are concerned, different airport lists may be provided for a given A/D/AD point, depending on whether the Arrival airport restriction list is defined or the Departure one or both of them at once.
3. no matter the shape of the 2D areas is but if a segment crosses several times the shape, it is not kept for assignment (local convexity check),
4. if a segment is crossing the area from outside the area and lying more than 10 NM outside the area at one side, it is not kept for assignment,
5. it is possible to define named E/X points that are not connected to any declared network. In that case, coordinates have to be put in the .frp file,
6. as well as Free route point file enable the automatic generation of route segments, it is possible to specify for A/D Free route points, a list of airports to enable automatic generation of segment usage restriction. More precisely, if some airport names follow the Free route point declaration in the .frp file, then any free route segment generated with this point will only be available for the listed airports,
7. parity of Free route segments is automatically assessed according to eastbound/westbound semicircular rule,
8. flight attitudes and Free route area entry/exit have to be consistent : e.g. if the end of a sequence of climbing segments goes to an Arrival exit point, or if the beginning of a sequence of descending segments lies at a Departure entry point, Free routing for these segments is disabled,
9. airports located closed to boundary of Free Route areas will possibly provide wrong results,

at least for flights filing RFL above FLxxx and having no time/space to leave Free Route Airspace before having reached the lowest allowed FL.

## Examples of Free Route cases

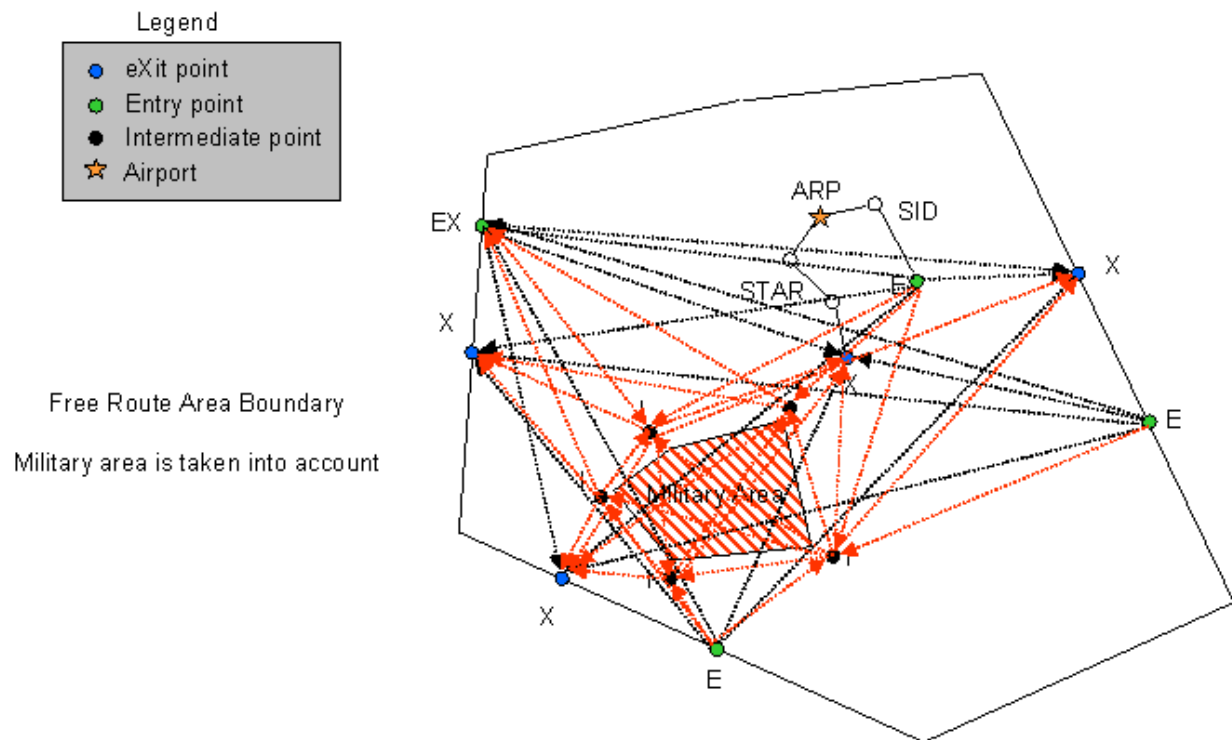
### Free Route examples with E/X/I points. Top view

Direct segments are generated between Entry to eXit. Intermediate points with given coordinates are not used. Direct segments are added to existing network, but not displayed. Assignment will generate the shortest path for each flight (no route options).



To take Military needs into account, during assignment, direct segments are generated between Entry to eXit, Entry to Intermediate, and Intermediate to eXit points, in order to generate a possible routing solution avoiding Military Areas. In such case the assignment will produce not one path per flight but several, from which Airspace Activation will choose the valid one, depending on Military time and level conditions. You will notice that direct segments generated with Intermediate points (in red) could cross Military area (useful if several Military areas are present in the Free Route Airspace). The generation of direct segments (red or black) is always added to the existing network (not displayed), enriching it.

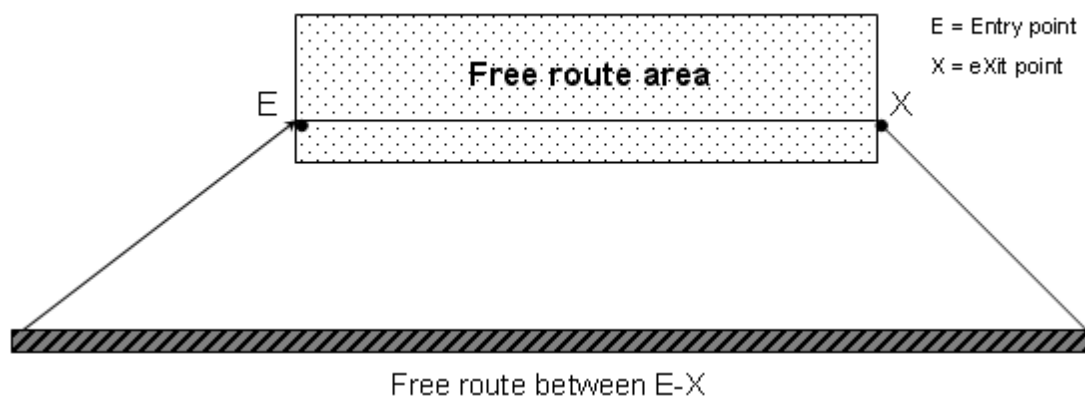


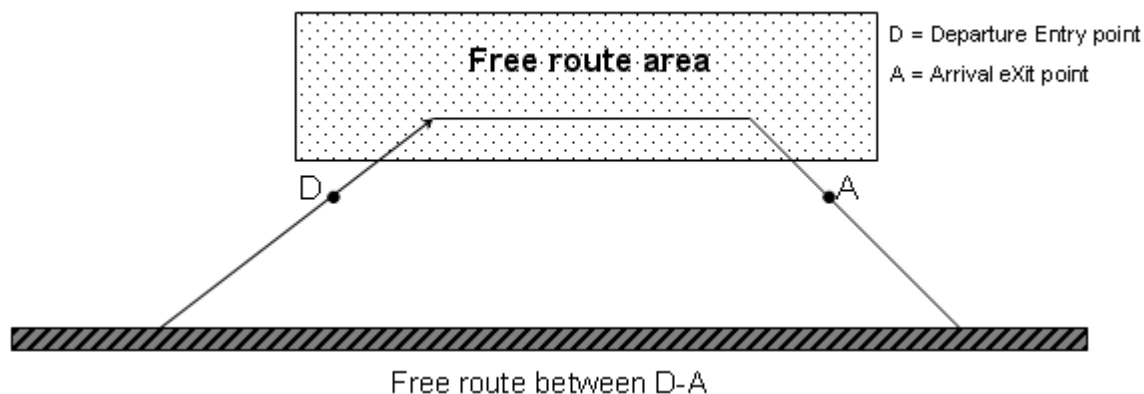


## Free Route examples with A/D points as well. Side view

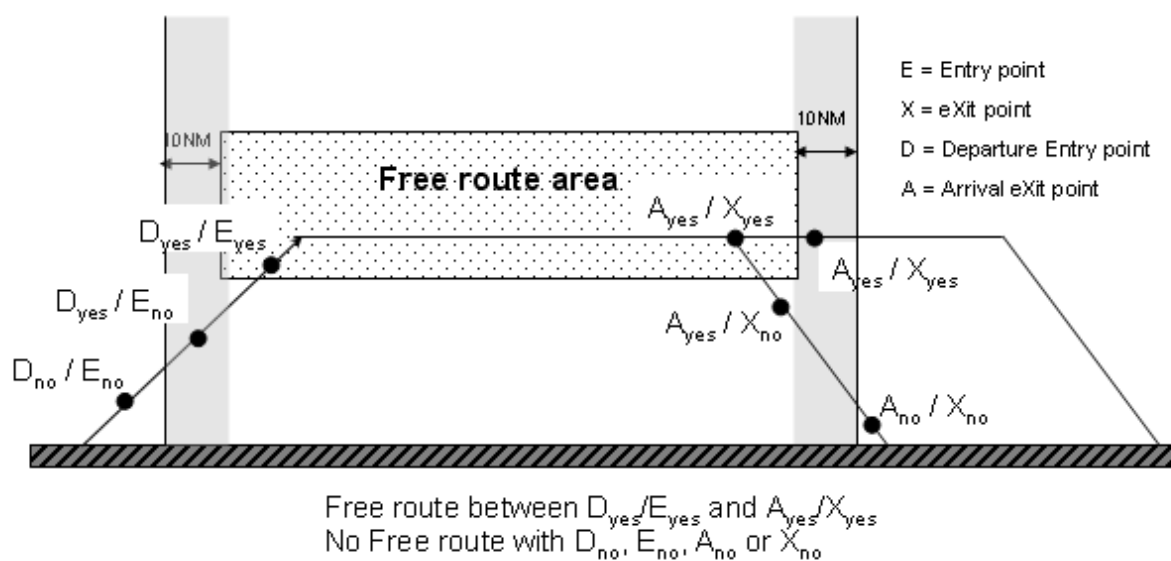
### 1) Basic cases

Requested Flight Level (RFL) is going through the Free Route area.

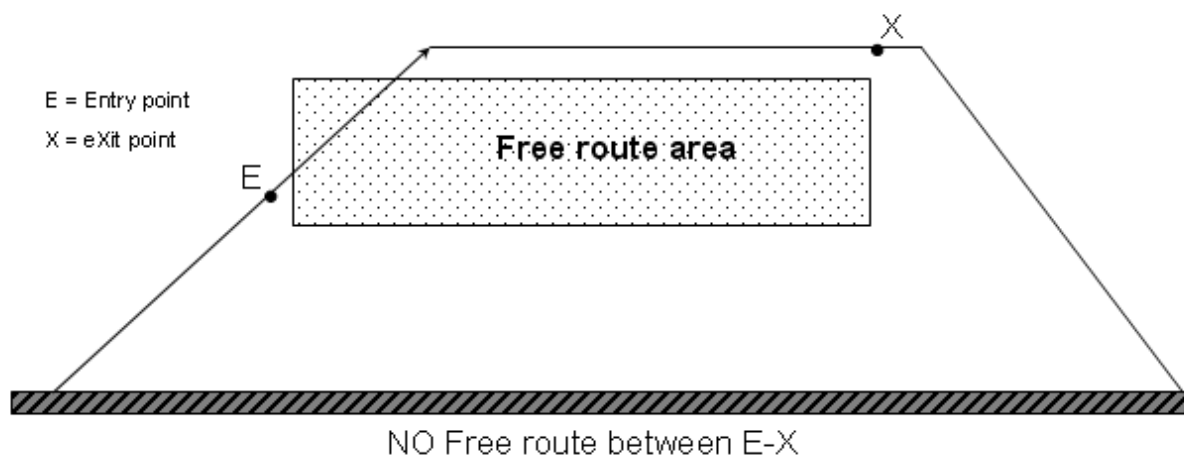




## 2) Generic cases

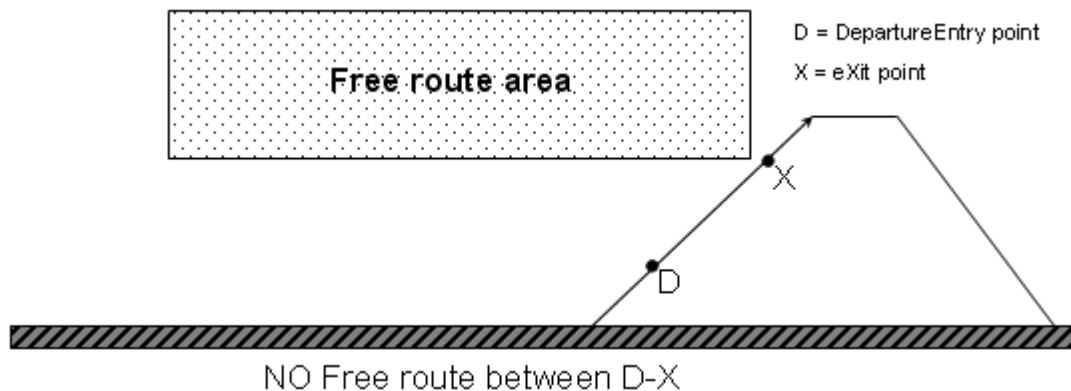


## 3) RFL above Free route area upper limit



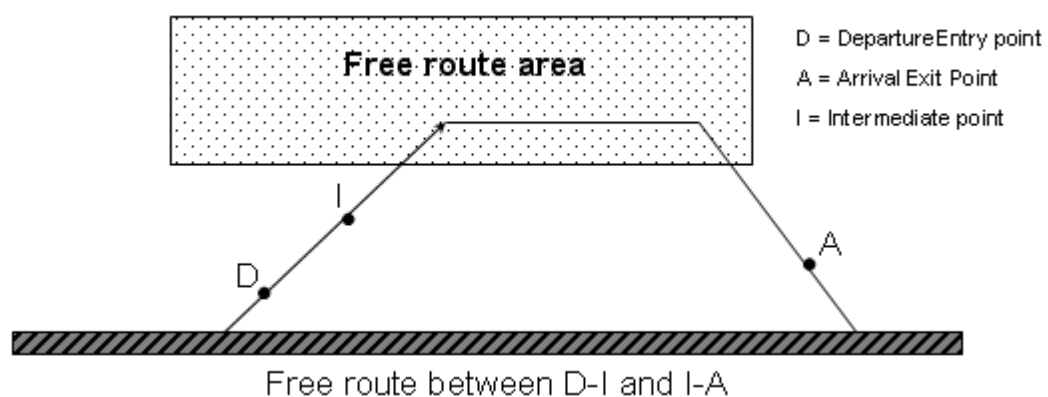
Requested Flight Level (RFL) is above the Free Route area.

#### 4) Departure point close to Free route area exit



The departure is passing an eXit point before Requested Flight Level (RFL) is reached .

#### 5) Free route with Intermediate point(s)

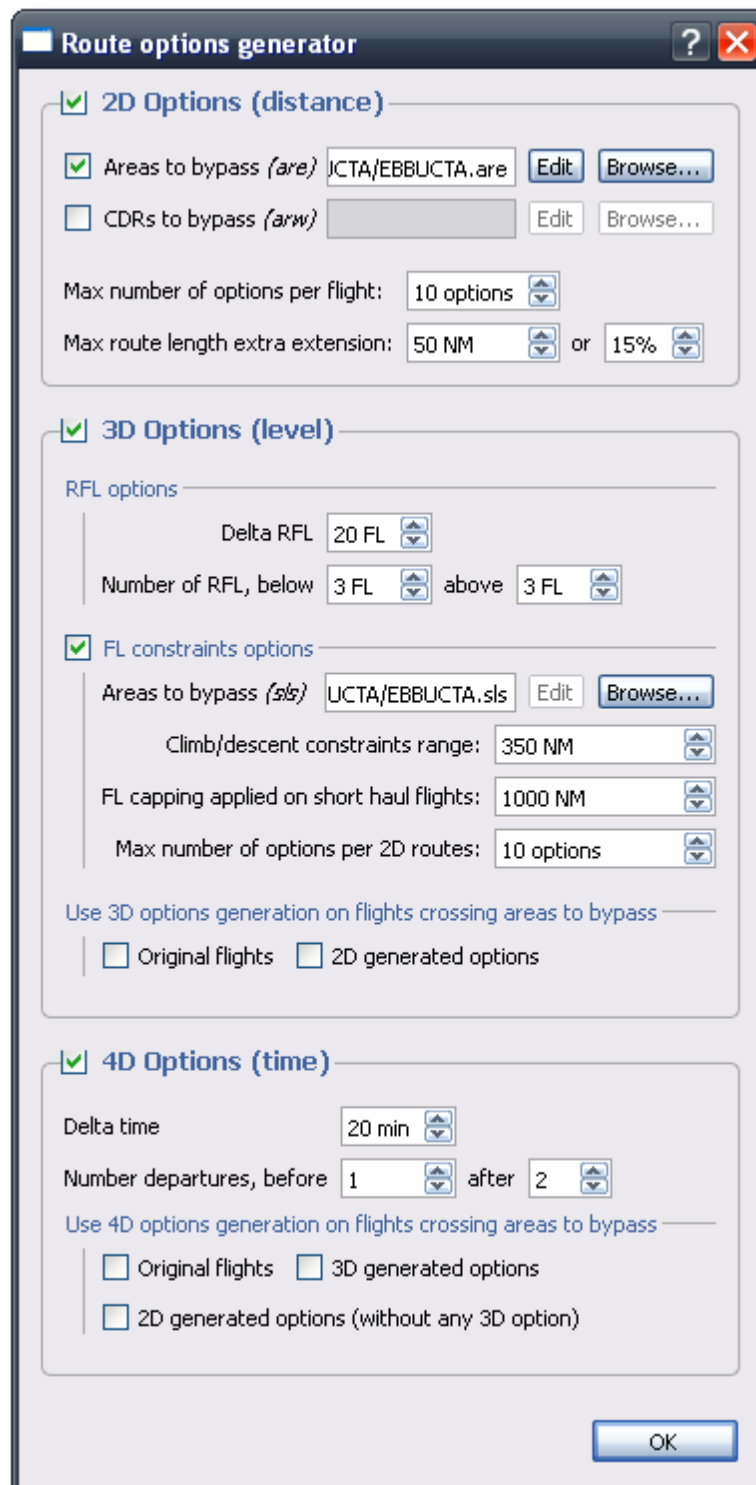


The departure is passing one or several Intermediate Points (I) before reaching Requested Flight Level (RFL).

### 2.4.1.3 Route Option Generator

#### Purpose / Description

Will find route options, in 2D and 3D, to avoid selected areas or CDRs. Several options per flight can be found, as long as they are within the set parameters. Can also generate options using the same route but with different RFL, 3D options, or using the same route and same RFL but with different departure times, 4D options.



**Route options generator**

☒ **2D Options (distance)**

☒ Areas to bypass (*are*)

☐ CDRs to bypass (*arw*)

Max number of options per flight:

Max route length extra extension:  or

☒ **3D Options (level)**

RFL options

Delta RFL

Number of RFL, below  above

☒ **FL constraints options**

Areas to bypass (*sls*)

Climb/descent constraints range:

FL capping applied on short haul flights:

Max number of options per 2D routes:

Use 3D options generation on flights crossing areas to bypass

☐ Original flights ☐ 2D generated options

☒ **4D Options (time)**

Delta time

Number departures, before  after

Use 4D options generation on flights crossing areas to bypass

☐ Original flights ☐ 3D generated options

☐ 2D generated options (without any 3D option)

**Route Option Generator Parameters Dialog Box**

## Parameters

All of these parameter files are text files which may be edited by the user.

- 1) **Enable 2D Options checkbox** – Check to activate **2D Options parameters** frame.  
**Areas to bypass** and **CDRs to bypass checkboxes and fields** – Use the browse

button to select a list of areas (.are file) and/or a list of CDRs (.arw file) to bypass. The route option generator will generate the shortest alternatives routes (taking into account penalisation) not going via any combination of these areas or CDRs. Format of .arw text file is : <airway\_name> <PT1-PT2> <PT2-PT3> ... with one airway definition per line.

**Max number of options per flight field** – To limit the number of options generated when the number of areas/CDRs to bypass is high.

**Max Route length extra extension fields** – The maximum extra route length extension of the alternative routes compared to the shortest route. The minimum of absolute extra extension and relative extra extension is considered as the maximum allowed extra extension.

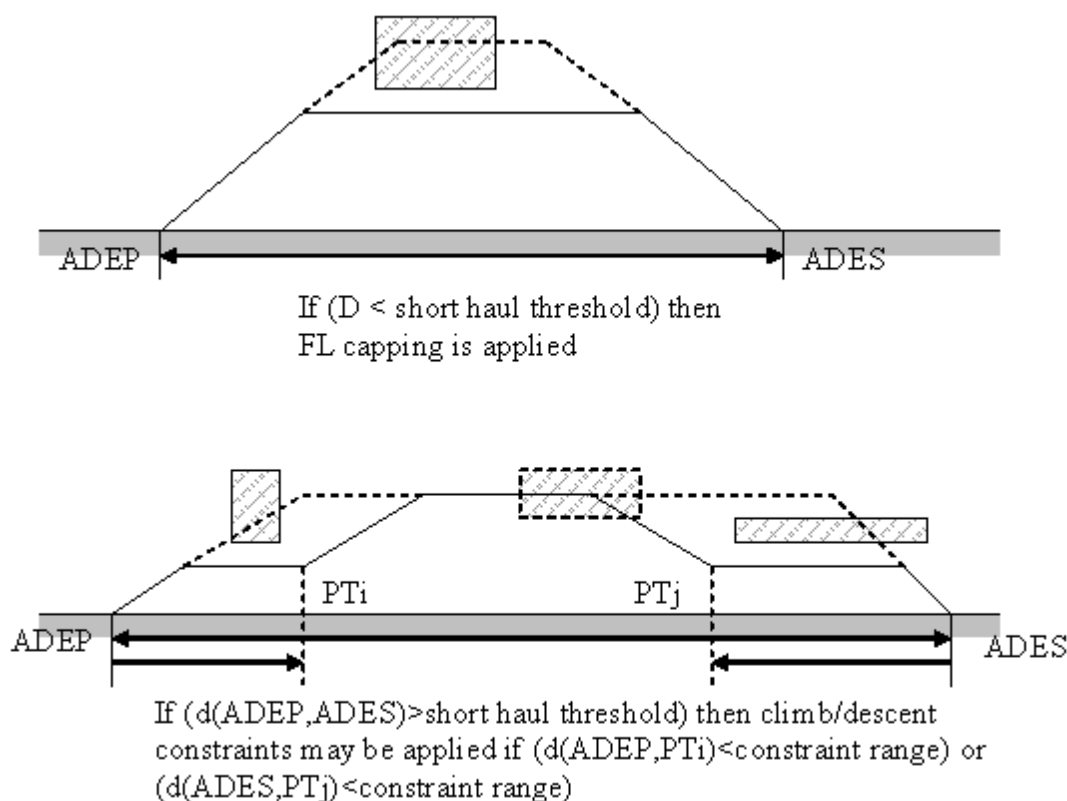
2) **Enable 3D Options checkbox** – Check to activate **3D Options parameters** frame.

**RFL options fields** – To generate options, with the same route but a different RFL (for later usage, in Profile calculation, see [Profile](#)), below or above the original RFL expressed in the expand file.

- **Delta RFL**
- **Number of RFL, below/above**

a) **FL constraints option frame** – To generate flight level constraints along the route in the produced ZIN file.

- **Areas to bypass checkbox and field** – To activate FL constraints option generation: the route option generator will produce alternative 3D routes (encoded in the ZIN file) going below the lower limit of sectors. For a given 2D route, a 3D route option will be generated for every combination of 3D lower levels of horizontally encountered sectors, even with vertically overlapping sectors.
- **Climb/descent constraints range and FL capping applied on short haul flights fields** – To specify the type of FL constraints (climb, descent, FL capping, refer to [3D Constraints](#).) See pictures below.



- c) **Use 3D options generation on flights crossing areas to bypass frame** – To generate 3D options on original 2D routes and/or 2D options.
- **original flights checkbox**
  - **2D generated options checkbox**
- 3) **Enable 4D Options checkbox** – Check to activate **4D Options parameters** frame.
- a. **Delta time and Number of departures fields** – To generate options, with the same 3D route but with a different departure time, before and/or after the original departure time expressed in expand file.
- b. **Use 4D options generations on flights crossing areas to bypass frame** – To generate 4D options on original 2D routes and/or 2D options and/or 3D options.
- **original flights checkbox**
  - **2D generated options checkbox**
  - **3D generated options checkbox**

## 2.4.2 Profile

### Accessed by

[Menu Bar / Processing / Profile](#)

### Purpose / Description

Profile generates from a 2D routes file (\*.zin) a 4D trajectory file (\*.so6), adding time and

flight level to each route point. This process support constraint data file, helping user to set departure and/or arrival and/or cruising flight level constraints for any flight or set of flights. The output \*.so6 file forms basic input for queries, airspace load and other key SAAM functions. The minimum cruising constraint that is supported is FL 30.

## See Also

- [2D Assignment process](#)
- [So6 to Direct Route PIN](#)

## Input

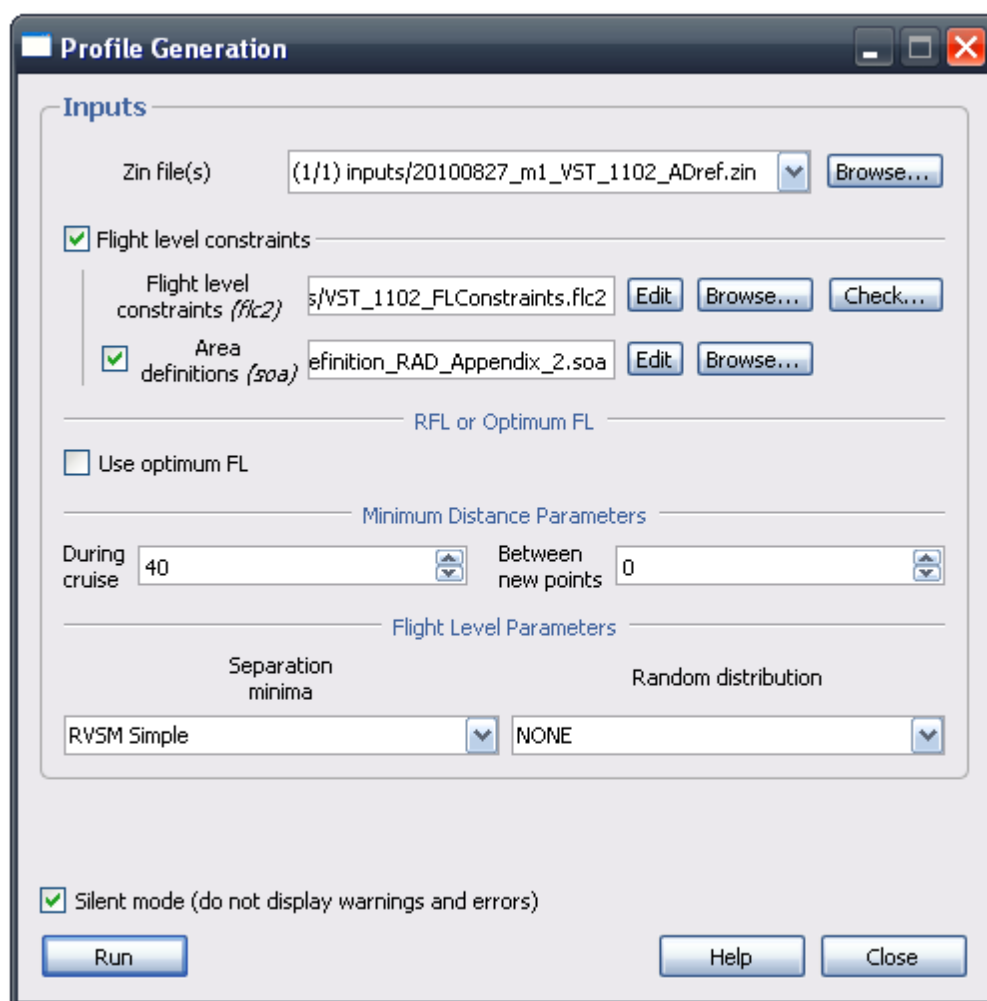
- [A 2D route description file \(\\*.pin or \\*.zin\)](#)
- [A constraint file \(\\*.flc2\) \(optional\)](#)
- An area definition file (\*.soa) (optional)

## Output

- A Traffic file (\*.so6)
- Diagnostic text files : constraint\_not\_respected.txt, missing\_actypes.txt, profile\_accurate.diag

## General Procedure

1. Specify the input files to be profiled
2. Specify a constraint file (*optional*)
3. Specify the area definition file (*optional*)
4. Set up the parameters (*optional*)
5. Run the Profiling process
6. When the process is completed three notepad windows will open if **Silent Mode** isn't checked.
  - 'Profile\_accurate.diag' contains flights that met conflicting constraints. The last line will tell you if all flights have been processed.
  - 'Constraint\_not\_respected.txt' contains constraints that could not be respected.
  - 'missing\_actypes.txt' contains flights for which the aircraft type was unknown in the performance tables. In that case, the generic B737 model is used instead.



**Profile Dialog Box**

## Parameters:

1. **PIN field** – Browse a \*.zin file, which is generated by the SAAM [Assignment process](#) or by [So6 to Direct Route PIN](#) . PIN file (Profile Input) contains airport origin/destination to all flights. ZIN file (Zone profile Input) contains zones defined in the assignment process. Instead of airport origin/destination airports are grouped into zones. The zone definition provides a facility for users to simplify the writing of profile constraints rules.
2. **Flight level constraints checkbox and field**– Tick the checkbox to select a constraint file (\*.flc2), which contains flight level constraints, by using the browse button.
  - i. **Check button** - Display an interface to check the consistency of the constraint file. See the [Check flight level constraint](#) dialog.
  - ii. **Area definition field** - Browse the area definition file associated to the constraint file (\*.soa).
3. **Use optimum FL checkbox** – Tick the checkbox if you want the aircraft to target their optimum flight level instead of their RFL as defined in the PIN file.
4. **Minimum Distance Parameters frame** –
5. **During Cruise field** - Indicates the minimum distance, in nautical miles, a flight has to fly on its cruising level. If the optimum profile cannot meet this condition then the cruise FL is automatically lowered. This parameter is ignored if the total flight length is shorter



than the set value.

6. **Between New Point** *field* - When calculating a profile, Top of Climb and Top of Descent points and other points linked to constraints are automatically generated by SAAM. To minimise the number of additional points that are created, the user may input a value, in nautical miles, for 'Between New Point'. SAAM will then look for an existing route point within the specified distance that could be used instead.
7. **Flight Level Parameters** *frame* -
  - i. **Separation Minima** *drop-down list* – Flight Level separation. Select CVSM or RVSM from the drop-down list. RVSM is default.
  - ii. **Random Distribution** *drop-down list* – Used to improve the degree of realism in profile calculations. For example, Flight Level allocation can be randomised to avoid having all equivalent flights (same City Pair, same aircraft) on the same level. This parameter can be set to 4 different levels (None, Narrow, Medium, Wide).

#### Randomisation Values

Type	Same FL	1FL Up	1FL Down	2FL Up	2FL Down	3FL Up	3FL Down
NARROW	90%	5%	5%				
MEDIUM	72%	10%	10%	4%	4%		
WIDE	50%	15%	15%	8%	8%	2%	2%

8. **Silent Mode** *checkbox* – Tick the checkbox if you don't want the notepad windows to be opened at the end of the process.

### 2.4.2.1 Check flight level constraints

#### Accessed by

The profile dialog box.

#### Purpose / Description

Check the syntax and the consistency of a constraint file (\*.flc2)

#### Input

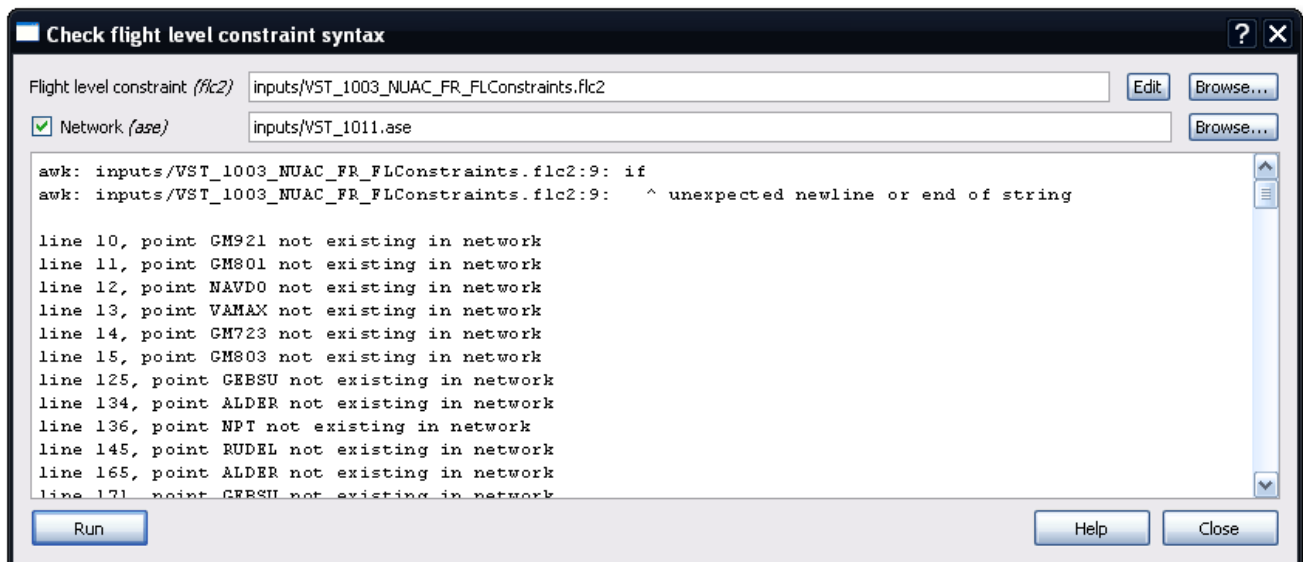
- [A constraint file \(\\*.flc2\)](#)
- A network file for the consistency check (\*.ase) (optional)

#### Output

- The errors are written in the text field of the dialog.

#### General Procedure

1. Specify Flight level constraint file.
2. Specify the network file (*optional*).
3. Click on the Run button.
4. The errors appears in the text box.



*The check flight level constraint dialog box*

## Parameters:

1. **Flight level constraint field** – Browse a \*.flc2 file.
2. **Network checkbox** – Browse a network file (\*.ase) in order to enable the consistency test. If the checkbox is unchecked, only the syntax check is performed.
3. **The text field** - Display the syntax and the consistency errors found in the constraint file. If there is no error, the text field is left blank.

## 2.4.3 Airspace/Traffic Intersection

### Accessed by

**Menu Bar / Processing / Airspace - Traffic Intersection**

### Purpose / Description

A process which computes 4D intersection of traffic with airspace volumes. The intersection, for the entry point and the exit point, is basically expressed with coordinates, flight level and time. Other fields are also processed like duration and distance within the airspace volumes. This information is stored in a file, by default named the same as the **.so6 traffic file, but with extension '.t5'**. This file is commonly used in many SAAM functions (query, airspace load, punching, work load etc.)

### See Also

- [Profile processing](#)
- SAAM [airspace editor](#)

## Input Requirement

- One or several 4D [SAAM Data files \(\\*.so6\)](#)
- [Airspace files](#), either in SAAM format or in Gasel format.
- T5 Cleaning Rules (\*.tcr) (*optional*)

## Output

- One or several Traffic Intersection files (\*.t5)

## General Procedure

1. Select one or several So6 files
2. The airspace files, either in SAAM format or in Gasel format.
3. Select a T5 Cleaning Rules file (\*.tcr) (*optional*)
4. Specify a scenario name for the output files (*optional*)
5. Click Run to start the process.

**Airspace/Traffic Intersection**

**Inputs**

Traffic (so6) (1/1) 20060518\_m1.so6 Browse...

Airspace Specification

Sectors (are or gar) 283/sansconf.are Browse...

Sectors (sls or gsl) 283/sansconf.sls Browse...

Optional parameters (Gasel file format)

☒ Airspace for collapsed sectors (spc) 283/Airspace.spc Browse...

☒ Configuration (cfg) 283/Configuration.cfg Browse...

☒ Opening scheme (cos) 283/OpeningScheme.cos Browse...

Other parameters

Minimum time spent in sector: 30 seconds

☐ T5 cleaning rules (tcr) Browse...

**Output**

Scenario name tst

T5 output file 20060518\_m1\_tst.t5

☒ Process with the new method  
(quicker, safer, but doesn't handle negative airspace)

Run Help Close

**Airspace/Traffic Intersection Dialog Box**

## Parameters

1. Traffic (so6) filename field - The traffic file(s) (\*.so6) may either be extracted from the CFMU database via SAMAD, or generated by SAAM itself through the [Assignment](#) and [Profile](#) processes. When selecting more than one \*.so6 file, the files have to be located in the same directory and are selected by standard windows file selection procedures (Shift or Ctrl + click). The number of files selected will be displayed in the \*.so6 filename field.
2. Airspace Specification frame – You can either use SAAM format or Gasel format for your sector files, this is done by ticking one of the radio buttons.

When “SAAM Sector file format” is selected you have browse two airspace files \*.are and \*.sls.

- a) Sectors (are) filename field – Airspace input file ‘\*.are’, is a file that contains geographical coordinates of boundaries with an associated name. In the \*.are file there can also be a lower and upper flight level connected to an area. If an associated \*.sls file is used, the level information in that will override the level information in the \*.are file.
- b) Sectors (sls) filename field - Airspace input file ‘\*.sls’, is a file that contains Airspace Volume building information. Each Airspace Volume has a name and consists of one or several airblocks. Airblocks can be added to an Airspace Volume, subtraction is deprecated, meta lines are not supported and cannot be used. An airblock is defined by a name and a specific lower and upper flight level. The coordinates for the airblocks are located in the associated \*.are file. The same airblock can be used by different Airspace Volume at different flight levels.

When “Gasel Sector file format” is selected you have browse two airspace files Airblock and Sector.

- a) Airblock filename field – This Gasel file contains geographical coordinates of Airblock boundaries (2D envelop).
- b) Sector filename field – This Gasel file contains information of how different airblocks are grouped together to build and create elementary sectors.
3. Optional parameters frame – More Gasel format files can be browsed and taken into account when running the Airspace/Traffic Intersection process
  - a) Collapsed sectors – Also called Airspace Gasel file. Where two or several elementary sectors can be grouped together into one sector, a collapsed sector. File name extension is \*.spc
  - b) Configuration – A set of elementary sectors and collapsed sectors used at a center at any time. Intersection is computed only for the sectors and collapsed sectors taking part in a described configuration. Airspace Gasel file, collapsed sectors, must be selected.
  - c) Opening Scheme - Period of time indicating when a configuration is operational. Intersection is computed taking into account the sector opening times described in the file. Requires Configuration and Airspace Gasel files. Not all configurations have an opening scheme.
4. Minimum time spent in sector (in seconds) field- A minimum time for the aircraft trajectories can be set here, the default value is 30 seconds, meaning that aircraft will only be counted if they stay more than 30 seconds within a sector. Shorter time values are not normally recommended as they are likely to produce less accurate results.
5. Clean T5 check-box and filename field – This is a tool to suppress flights from an airspace. The rules for this are stored in a file called ‘T5 Cleaning Rules’ (\*.tcr), managed through the Query dialog box (see 5.6.1.2).
6. Output frame – The output file will be given the same name as the traffic input file (\*.so6), and will be automatically recognised in most of the SAAM dialog boxes.
  - a) Scenario name field – Sometimes it is necessary to give the output file a different

name. This is done by adding a suffix to the output name. The output file will be 'nameso6\_suffix.t5'.

- b) T5 output file field – Will show the name of the output file/s.
- 7. Process with new method (quicker, safer, but doesn't handle negative airspace) check-box
  - SAAM is gradually getting rid off the negative airspace, to make SAAM more compatible with other tools.

## 2.4.4 Free Route

### Accessed by

**Menu Bar / Processing / Free Route**

### Purpose / Description

The **Free Route** option calculates an intermediate file (\*.pin), used for profile calculation, with a straight trajectory between entry and exit points for a particular Free Route Airspace.

WARNING : this module is a deprecated version of Free Route processing in SAAM since a better way of dealing with Free Route airspace structure is available within Assignment processing (see [Free route in assignment](#)).

### See Also

- [Airspace/Traffic Intersection](#)
- [Profile processing](#)

### Input

- Traffic file (\*.so6)
- Airspace / Traffic Intersection file (\*.t5)
- Entry / Exit Point file (\*.eep) (*optional*)

### Output

- an intermediate file (\*.pin) for profile calculation
- Flight Level Constraint file (\*\_FR.flc2) (*optional*)

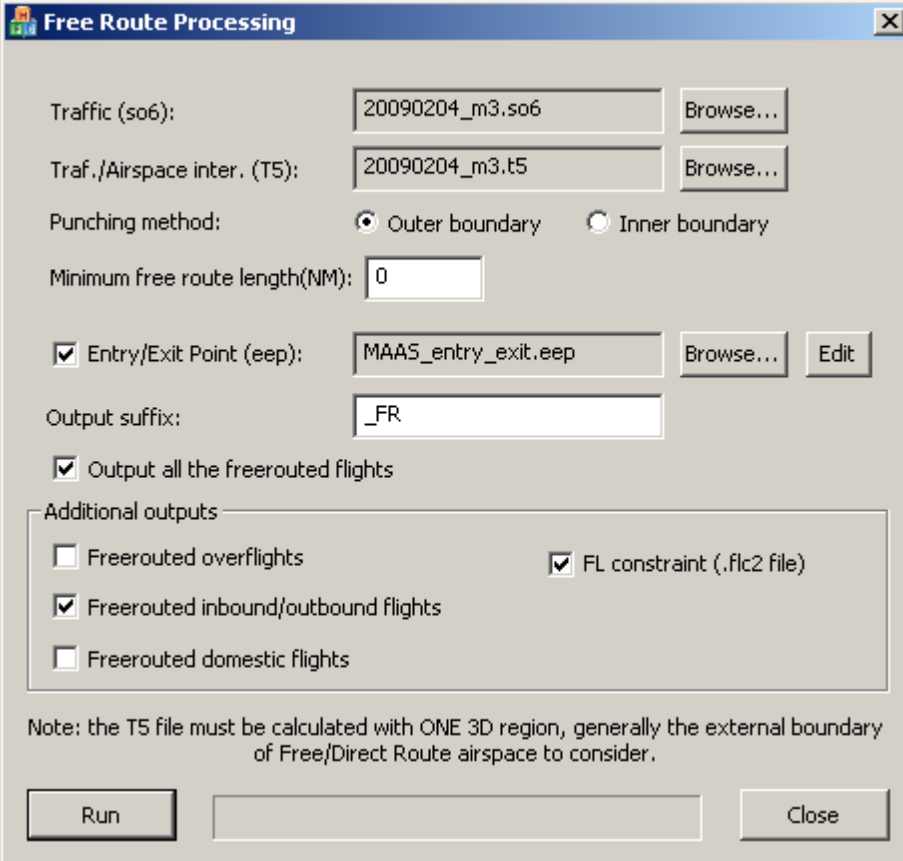
### General Procedure

First a \*.t5 file has to be generated using the Free Route Airspace ( \*.are and \*.sls) you are interested in and a traffic file (\*.so6). For this you use the [Airspace/Traffic Intersection](#) process.

In the Free Route Process you use the original traffic file (\*.so6), same file you used in the Airspace/Traffic Intersection process and the Airspace/Traffic Intersection output file (\*.t5) to generate an intermediate file (\*.pin) you then use for profile calculation.

The Free Route process calculates a straight trajectory between entry and exit point for the Free Route Airspace. The default is, if no Entry/Exit Point file is selected, the last point before entering the Free Route airspace is considered to be entry point and the first point after leaving the airspace is exit point. If an Entry/Exit Point file is selected, then each flight is

checked to see if any constraints or changes written in the Entry/Exit Point file can be applied to this flight, if not it will follow the default.



The dialog box is titled "Free Route Processing". It contains the following fields and controls:

- Traffic (so6):** Text field with "20090204\_m3.so6" and a "Browse..." button.
- Traf./Airspace inter. (T5):** Text field with "20090204\_m3.t5" and a "Browse..." button.
- Punching method:** Radio buttons for "Outer boundary" (selected) and "Inner boundary".
- Minimum free route length(NM):** Text field with "0".
- Entry/Exit Point (eep):** Checked checkbox, text field with "MAAS\_entry\_exit.eep", "Browse..." button, and "Edit" button.
- Output suffix:** Text field with "\_FR".
- Output all the freerouted flights:** Checked checkbox.
- Additional outputs:** A group box containing:
  - Freerouted overflights:** Unchecked checkbox.
  - Freerouted inbound/outbound flights:** Checked checkbox.
  - Freerouted domestic flights:** Unchecked checkbox.
  - FL constraint (.flc2 file):** Checked checkbox.
- Note:** the T5 file must be calculated with ONE 3D region, generally the external boundary of Free/Direct Route airspace to consider.
- Run:** Button.
- Close:** Button.

**Free Route Processing dialog box**

## Notes

1. **Traffic (so6) filename field** - The traffic file (\*.so6) may either be extracted from the CFMU database via SAMAD, or generated by SAAM itself through the assignment and profile processes.
2. **Traf./Airspace inter. (T5) filename field** - Traffic/Airspace Intersection file (\*.t5) generated by the Traffic/Airspace Intersection process.
3. **Minimum free route length (NM) field** - To set a minimum free route distance. Free route is deactivated if distance between entry and exit point is below the specified threshold.
4. **Entry/Exit Point (eep) filename field** - Tick the check-box to activate the filename field and browse. This file helps to organise the free route in the specified area, by forcing the traffic to enter or exit the area via predefined points.
5. **Output Suffix field** - A prefix that will be given to all output files. By default it is "\_FR", but this can be changed.
6. **Output all the freerouted flights check-box** - When ticked will output all free routed flights
7. **Additional outputs frame** - Gives the user the possibility to output flights in separate files: overflights, inbound/outbound and domestic, or to generate flight level constraints.
  - a. **Freerouted overflights check-box**

- b. **Freerouted inbound/outbound flights check-box**
- c. **Freerouted domestic flights check-box** –
- d. **FL constraint (flc2 file) check-box** – When ticked will generated a file that contains flight level constraints found in the input traffic file (\*.so6). These constraints can then be re-injected when running the profile calculation.

## 2.4.5 Forecast

### Accessed by

**Menu Bar / Processing / Forecast**

### Purpose / Description

Generates a forecast for future traffic demand using 'STATFOR' data file (\*.sff).

### Input Requirement

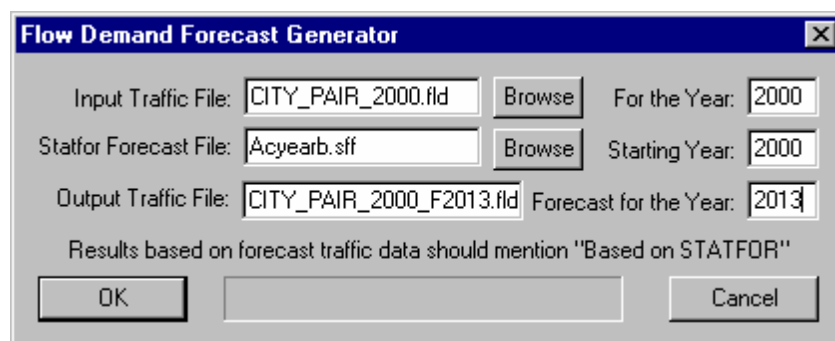
- A flight demand traffic file (\*.fld)
- A Statfor Forecast file (\*.sff)
- A Statfor zone definition called "FORECAST\_DEF\_ZONE.txt"

### Output

- A forecasted Output Traffic File (\*.fld)

### General Procedure

- Select input traffic file (\*.fld) and Statfor Forecast file (\*.sff) using the Browse buttons
- Specify a name for the output file.
- Specify the year from which the input Traffic file was issued
- Specify the first year for which the Statfor Forecast file was prepared
- Specify the year for which you want to generate forecast
- Click OK to run the process.



**Flow Demand Forecast Generator Dialog Box**

## Note

1. Input Traffic File field – Browse a flight demand traffic file (\*.fld)
2. Statfor Forecast File field - Can be obtained from the STATFOR department (\*.sff).
3. Output Traffic File field - will be a \*.fld file containing the name of the target year. This \*.fld format can be used to process densities and forecasted T5 files. The output file name is created by Input Traffic File name plus “\_FYYYY”. “F” means Forecasted, and “YYYY” is the year for which you want to generate a forecast.

## 2.4.6 Optimisation

Not yet implemented

## 2.4.7 Route Choice Based on Military Opening Time

### Accessed by

[Menu Bar > Processing > Route Choice Based on Military Opening Time](#)

### Purpose / Description

After running [Route Option Generator](#) and profile calculation, you can choose the cheapest available route depending whether a military area is opened or closed (this information is stored in an editable file called \*.mot).

Cheapest choice means: the cost of a route option is today based on time required to fly the route from origin to destination, including RAD, RFL and FL constraints. SAAM choose the cheapest route if the the original shortest route is not valid, else this last is always taken. This behaviour tries to keep stable solution if possible (avoid to change the shortest choice just because of Route Option generation).

### Input

- Traffic file (\*.so6)
- Option file (\*.lox)
- Region file (\*.are or \*.gar)
- Sector List file file (\*.sls or \*.gsl)
- Airspace Activation Time file (\*.mot) - Only the MOT format 2 (time and level) is supported.

### Output

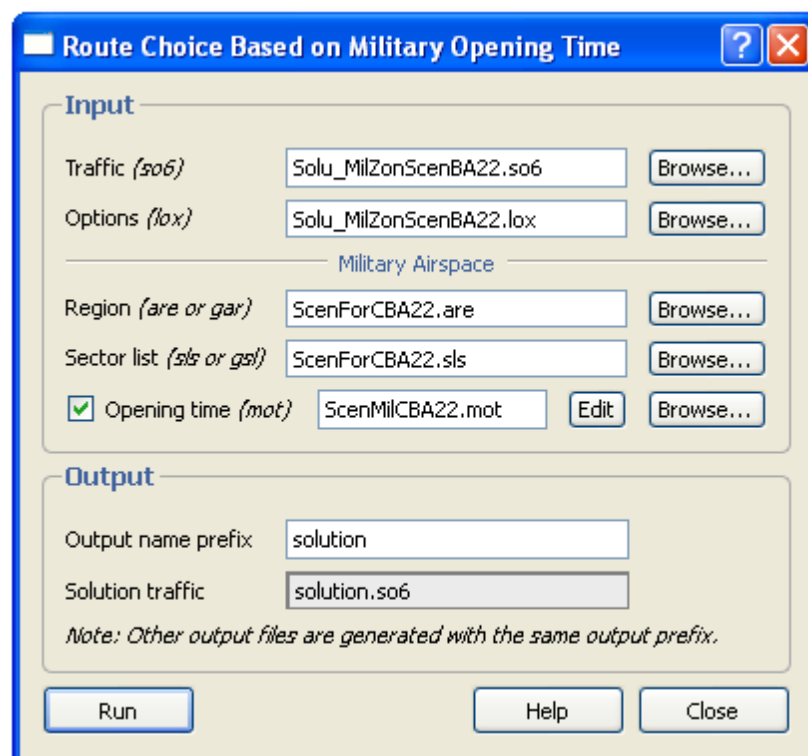
- solution\_HasOption\_err.so6 – Contains flights where a route option is found, but the option is no good.
- solution\_NoOption\_err.so6 – Contains all flights where a route option wasn't found
- solution\_OnlyOptionsWithRef.so6 – Contains flights for which an optional route was chosen, with it's original route.



- `solution_original_unchanged.so6` – Contains flights for which the chosen optional or original route is the same as the original one.
- `solution_rerouted.so6` – Contains flights for which the chosen optional or original route is different from the original one.
- `solution_rerouted_original.so6` – Contains the original flights for which the chosen optional or original route is different from the original one.
- `solution_to_be_regulated.so6` – Contains flights that are not part of the solution and for which no correct option could be found.
- `solution_anim.tdv` – An animation is built with traffic from `solution.so6` and the military areas. The military areas are displayed open (green) or closed (red) depending on Military Opening Time.
- `solution.so6` – Contains all flights. The original flights that didn't need a route option and the flights for which an optional route was chosen.
- `solution.lox` - If a copy of the input `*.lox` file on which the flags "validity" (field #3) and "choice" (field #4) are set depending on route availability and choice that is made. Useful to understand why a route was chosen.
- `solution_compare_rerouted.txt` contains a table comparing original flights with their chosen rerouted options. They are sorted according to their longest route length difference.

## General Procedure

1. Browse the requested input files.
2. Type an Output name prefix or accept the automatic "solution"
3. Click Run



### **Military Opening Time Resolution Dialog Box**

## **Parameters**

1. **Input frame** – Browse all the requested input files generated by the Route Option Generator process.
  - a) **Traffic file** (\*.so6) *field*
  - b) **Option file** (\*.lox) *field*
2. **Military Airspace frame**
  - a) **Region file** (.are | .gar) *field*
  - b) **Sector list file** (.sls | .gls) *field*
  - c) **Opening Time file** (\*.mot) *field* - *Optional* - Browse a mot file. Clicking on the **Edit** button opens the [Airspace Activation Editor](#). *Note:* You can only edit MOT(s) in format 2 (time and level).
3. **Output frame** –
  - a) **Output name prefix** *field* – Type an Output name prefix. This prefix will be given to all the generated output files. You can also accept the default one “solution”.
  - b) **Solution file** *field* – The name of the output traffic file (\*.so6), depending on prefix, is displayed here.

## **2.4.8 Cheapest Route**

### **Accessed by**

**Menu Bar / Processing / Cheapest Route Choice**

### **Purpose / Description**

This process will produce two SAAM \*.so6 traffic data: one containing the cheapest routings, the other one the shortest routings. The number of flights in these files will be equal to the number of flights found in the original CFMU Flight Plan Data Traffic file (possible missing flights coming from assignment are re-injected).

### **Input**

- The original Flight Plan Traffic data file (generally coming from CFMU via [SAAM Traffic Data Generator](#)), it is used to automatically re-inject missing flights.
- The 4D assigned traffic file generated containing Route Options
- The cost in euros for flying one NM (by default set to 15 euros).

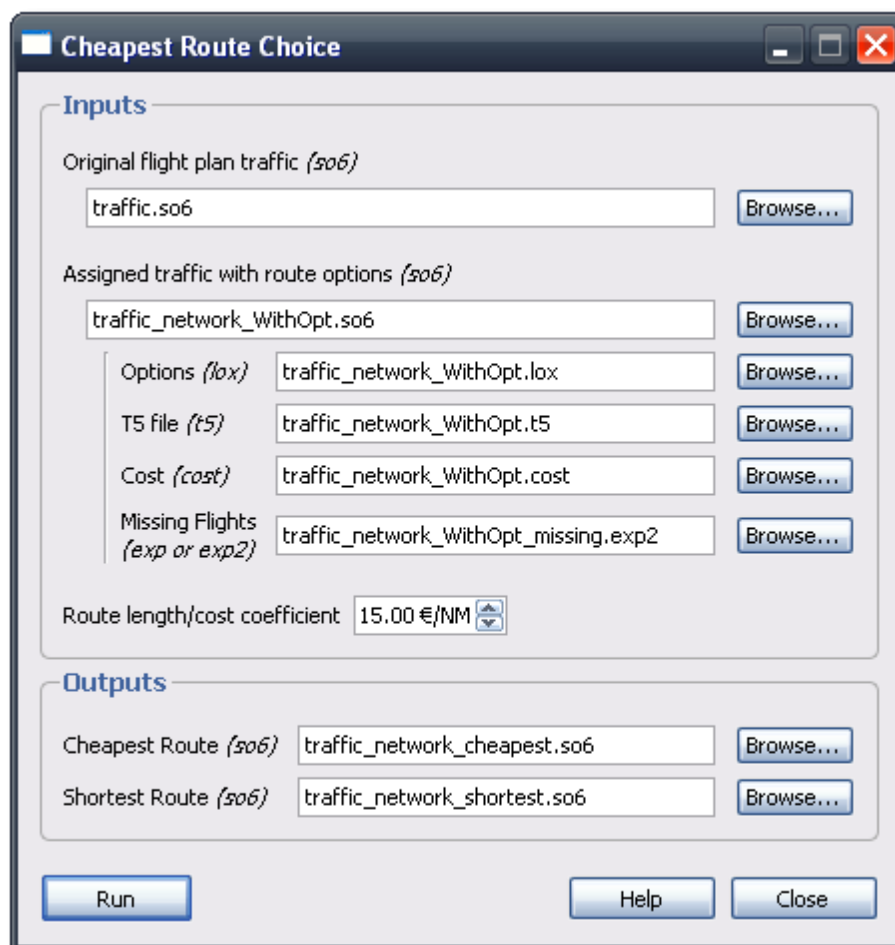
*Note:* This process will also read associated other files: \*.lox (produce by assignment with 2D Route Option Generation turn ON), \*.t5 (produced by Airspace/Traffic intersection of 4D traffic with Route Option and Route Charge Areas), \*.cost (produced by [Route Charge module](#)), they all that must exist in the same working directory.

### **Output**

- Two files: name\_cheapest.so6 and name\_shortest.so6 with name being the name of your input 4D assigned traffic file.

## General Procedure

1. Import a CFMU Flight Plan traffic data file (menu Import / Data from [Intranet](#) or [Extranet](#))
2. Create a traffic demand file (\*.exp file) (menu [Transform So6 to.../So6 to Traffic Demand](#))
3. Import the set of files for Route Charge processing (menu [Import / Get Basic Airspace Data / Route Charge Set](#))
4. Launch an assignment with 2D Route Option turned ON using Route Charge Area file (\*.are file) previously imported at step 3 (menu [Processing/Assignment](#))
5. Launch profile calculation (menu [Processing/Profile](#)) with file coming out from step 4
6. Launch Airspace/Traffic intersection of 4D traffic with Options coming out from Profile calculation (step 5) with Route Charge Areas (\*.are and \*.sls previously imported at step 3) (menu [Processing Airspace/Traffic Intersection](#)),
7. Launch Route Charge Calculation (menu [Analysis/Route Charge](#) with cost check box ON and possibly 20Km adjustment) with input files coming from previous steps. You have here the freedom to change the Unit Rate, but take care that fake areas where added (RV, RW, RX,RY) to fine tune the behaviour of the cheapest route.
8. Launch Cheapest Route Process (menu [Process / Cheapest Route Choice](#)). The first input (Original Flight Plan Traffic) must be the traffic file imported during step 1, while the second input (Assigned Traffic with Route options) must be the traffic file coming out from step 5.



### *The Cheapest Route Dialog Box*

## Parameters

### 1. Inputs frame -

- a) **Original flight plan traffic (so6) field** – Browse the original traffic file (\*.so6)
- b) **Assigned traffic with route options (so6) field** – Browse the assigned traffic file (\*.so6). If every previous processes have been run, the associated files will be automatically selected.
- c) **Option (lox) field** – Browse the associated option file (\*.lox)
- d) **T5 file (t5) field** – Browse the associated airspace intersection file (\*.t5)
- e) **Cost (cost) field** – Browse the associated cost file (\*.cost)
- f) **Missing Flights (exp) field** – Optional - Browse the associated missing expand file, (\*\_missing.exp). This file is optional, the process can run without it.
- g) **Route length/cosy coefficient value** - Select the cost of one nautical mile. The default value is 15€/NM.

### 2. Outputs frame -

- a) **Cheapest Route (so6) field** - Select the name of the cheapest output file (\*.so6)
- b) **Shortest Route (so6) field** - Select the name of the shortest output file (\*.so6)

## 2.4.9 SIM

### Accessed by

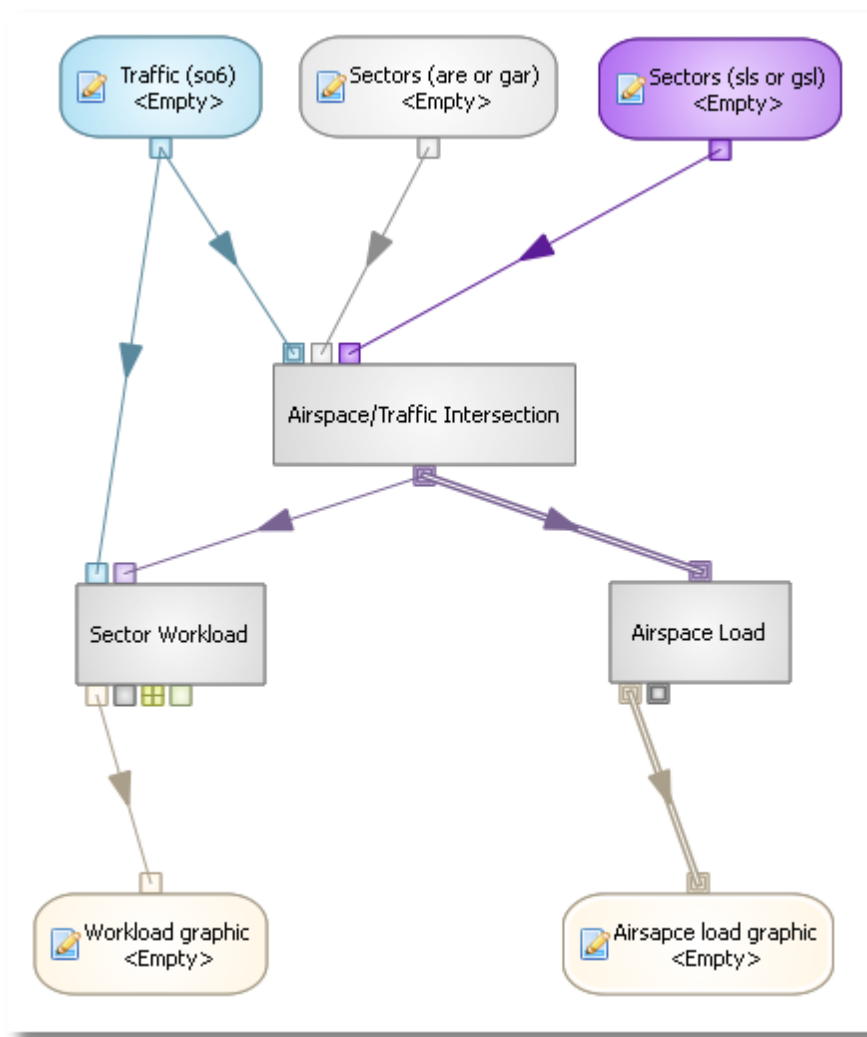
[Menu Bar / Processing / SIM / Launch Designer](#)

or by the  button on the toolbar.

### Purpose / Description

SAAM Intuitive Maker (SIM) is a software tool integrated in SAAM and aimed at the automation of complex, repetitive and error-prone SAAM tasks. These tasks are usually composed of multiple interactions with various SAAM dialog boxes, branched together by their input and output.

The main idea behind SIM is to represent these SAAM tasks in an efficient and easily readable way: the SIM diagrams. The goal for these diagrams is to be editable, configurable, executable and reusable.



A simple SIM diagram.

It runs the "Airspace/Traffic intersection" module, then run two analysis: "Sector workload" and "Airspace load".

A SIM diagram is composed of data blocks (inputs/outputs) and processes which are linked together to create a flow. When you run a diagram, its processes are executed in correct order and the transmission of data is done automatically. See the [diagram components](#) page to learn more about the composition of a diagram.

## Intended users

Designing your own diagrams is a very complex process and it takes an advanced user with some technical knowledge to manage it.

The [extension guidelines](#) will provide more information about the possibilities of SIM. It is intended for users with a very good technical knowledge.

For standard users [SIM](#) has several built-in SIM diagrams provided by SAAM via the **RUN** command (F8 key or **Menu Bar / Processing / SIM / Run Diagram...**). They are provided to help users to run common work flows automatically.

## Run built-in SIM diagrams

Under the **RUN** command (F8 key or **Menu Bar / Processing / SIM / Run Diagram...**)

several built-in SIM diagrams can be found . When selecting and opening a diagram a dialog box will appear, where all necessary input files must be browsed. Then start the execution.

### Built-in SIM diagrams

Name of diagram	Processes executed in following order
Complete Scenario Run	So6 To Expand Assignment Profile Matching Combine Files Airspace/Traffic Intersection
Simple Complete Scenario Run	Same processes as in "Complete Scenario Run". The difference is that there are less input files to browse. The environment is automatically retrieved at the location of the VST file.

## 2.4.10 Generic Launch

### Accessed by

**Menu Bar / Processing /Generic Launch**

### Purpose / Description

Generic Launch is a facility that allows the user to run any program using the SAAM Interface. Programs can be either executable (\*.exe) or Korn Shell (\*.ksh) files.

### Input

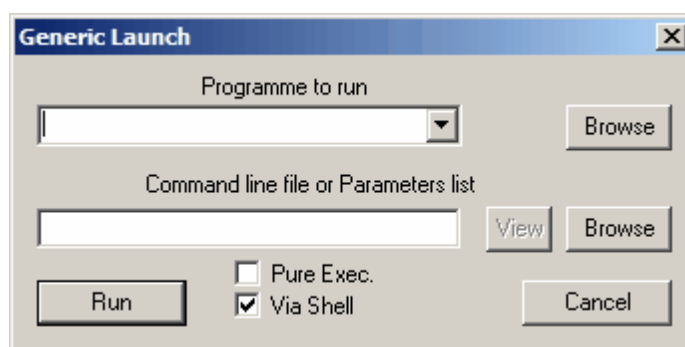
- An executable file
- One or several optional parameter values

### Output

- Depends on the function of the input executable file

### General Procedure

1. Ensure that the executable file/s is located in the SAAM/ bin directory
2. Load files into the input fields using the Browse buttons
3. Type in parameters (*optional*)
4. Click Run



**Generic Launch Dialog Box**

## Parameters

1. **Program to run filename field** – By clicking on the arrow of the combo box, you will get a set of predefined executable or korn shell. This list is located in SAAM / bin in the file “generic\_launch\_list.txt”. The file can be updated by users, but might be lost if a newer version of SAAM is loaded and used. By default this list of pre-defined programmes contains:
  - DLGB\_LessThanTime.exe is a dialog box proposing to extract flights (in the form of so6 and t5 file) spending less than N time in sectors.
  - ColorTDV2.exe is a dialog box to create a TDV with coloured sectors, the colour being based on thresholds and any input data, allowing a lot of possibilities. Warning: the sectors to be displayed needs to be transformed once into TDV before (menu transform / Airspace to TDV).
  - cruiseFLselect.ksh needs 3 parameters: input\_traffic.so6 file, min FL, max FL. It will extract from input traffic file all flights having max cruise level between the specified limits. Output goes to console so a redirection into a proper so6 file is required. Example: of parameters to be typed: 080910\_m1.so6 245 285 > 080910\_m1\_FL245\_FL285.so6
2. **Command line file or Parameter list filename field** – One or several optional parameter values can be typed in the parameter input field. These parameters will be given to the program when it's launched. Because it is difficult to remember by heart the list and type of parameter(s) for a given executable or korn shell, a facility is provided with a '\*.pal' file. This file must be located in your working directory and contains on the first line the list of parameters for a given executable or korn shell file. When browsed, the content of the file is displayed in the parameter input field. To change the value of the parameter, it is more convenient to click on button 'view'.
3. Generic launch automatically detects which type of executable is used, executable or korn shell, and checks the corresponding box.
  - **Pure Exec checkbox** – For executable programs
  - **Via Shell checkbox** – For korn shell programs
4. **Run button** – Launches chosen program

## 2.4.11 SAAM Bash Console

### Accessed by

[Menu Bar > Processing > SAAM Bash Console](#)

### Purpose / Description

To give users an easy access to a shell console, to run script files and other shell commands. The shell console will always be opened by default in your working directory (you can change it with "cd" command).

There is a limited number of commands that will be recognized:

- cygwin shell commands
- alias created by SAAM
- commands found in "SAAM/bin" (either shell scripts or exe files) but also, if not found in SAAM/bin it will search the command in your working directory (which is convenient when users want to write/experiment their own script or exe without changing the content of SAAM/bin)

```

C:\SAAM_V3_3_2_06nov07\bin\gnu\bash.exe
[c:/] ls
8.33kHz          MSDOS.SYS        SAAM_files
AClient.cfg      MSOCache         Spel
AClient.dat      Module_1         System Volume Information
AMCleanUp.log    My_MKS           IAAM_ref_manual
AUTOEXEC.BAT    NEUAC 1.2        TEST
Annika           NTDETECT.COM     Test_Estonia
CD_cover        New_ACC          Traffic_Data_per_State
CFMU_ACC        OneSky_Teams     UST_data
CFMU_e-learning Ora817           Vincent
CONFIG.SYS      Program Files    WINDOWS
Com_2020        RECYCLER         Web_Training_Data
Config.Msi      SAAM             boot.ini
Documents and Settings SAAM_Extranet   hiberfil.sys
ENAV           SAAM_Flash       ntldr
Eurocontrol     SAAM_Internet    pagefile.sys
IO.SYS          SAAM_Leaflet     saam_default_scenario
IUTemp         SAAM_Training_test screenhunter
Luca_Bellesia  SAAM_USER_PREF
MLL_Sky        SAAM_U3_3_2_06nov07

[c:/] cd Traffic_Data_per_State
[c:/Traffic_Data_per_State] ls
Albania Croatia Estonia Poland
Austria Czeckien Lithuania Serbia_Montenegro
[c:/Traffic_Data_per_State]
  
```

**SAAM Bash Console**



### Some classical Cygwin Commands (some can be advantageously grouped with | pipeline)

Command	Description
<code>cd directory</code>	cd (like change directory) allows to move away from your current default working directory
<code>rm file</code>	rm (like remove) delete the file
<code>cp file1 file2</code>	copy file1 into file2
<code>mv file1 file2</code>	move file1 to file2
<code>touch file</code>	update time stamp of the file if file exist else create the file
<code>head file</code>	displays the 20 first lines of file
<code>tail file</code>	display the 20 last lines of file
<code>grep pattern file</code>	display all lines of file that contains the pattern
<code>wc -l file</code>	counts the number of lines of file
<code>sort file</code>	sort file lines (see -k option to sort of some rows)
<code>uniq -c file</code>	if file is sorted, group same lines together and display the unique candidate with associated number of grouped lines
<code>awk '{your processing}' file</code>	powerful interpreter, using 'C' like syntax, automatically processing file line by line

### Some practical SAAM alias Commands

Alias	Description
<code>n</code>	Calling a text editor (by default notepad2.exe). If the text file to edit does not exist, it is first created then opened with the text editor. If n is typed without parameters then the file "blank.txt" is first created and then edited. This is very useful to view/change any text file (results, input/output files, parameters, Bash shell files ...)  <code>n contPlnVST08_02.flc2 n</code> → will edit flight level constraint file
<code>lss</code>	(s like size) lists the 10 biggest files from your current working directory
<code>lst</code>	(t like time) lists the 10 most recent files from your current working directory
<code>h</code>	h (like history) lists the 10 last commands you have typed and run (note: commands can be recalled with up/down keyboard arrows, and possibly edited with left/right keyboard arrows).
<code>d</code>	d (like delta) launch a tool for graphical text file comparison "Kdiff3.exe". If no parameter are given user will browse files to be compared from Kdiff3.exe user interface.  <code>d contPlnVST07_1.flc2 contPlnVST08_02.flc2</code> → To compare changes in Flight Level Constraint data files between one year of VST (assuming the 2 or 3 files to be compared are present in your current working directory, otherwise put the correct path).

## Some examples of some SAAM "exe" programs

Name	Description
<a href="#">Coordinates.exe</a>	Convert columns of minute decimal coordinates (lat & lon) from an input file into string coordinates (and vice versa)
<a href="#">NetworkCoordCheck.exe</a>	Check coordinates location of points of a network (*.ase) against point coordinates provided in a *.navpoint file. Several output are possible, in particular a "cleaned" network.
<a href="#">RemoveTMARoutePart.exe</a>	Used to create a direct path within the TMA part of input traffic SO6 file (based on given TMA radix). The output is an SO6 file.
<a href="#">newPointFromAzimLength.exe</a>	<p>Used to create new coordinates from:</p> <ul style="list-style-type: none"> <li>* coordinates (in SAAM minutes decimal format),</li> <li>* an azimuth (in degrees),</li> <li>* a length (in NM).</li> </ul> <p>Like the other tools in this category, it requires as input a set of columns position that will be used to fetch the necessary information in the input data file or from the command line (manual input).</p> <p>Example: what are the coordinates of a point located at 850NM, 90 degrees azimuth from coordinates lat 2700 and lon 0 ?</p> <pre>newPointFromAzimLength.exe 1 2 3 4 2700 0 90 850 <b>2596.979012 1178.719894</b> 2700 0 90 850</pre>
<a href="#">newPointFromCoordLength.exe</a>	<p>Used to create SAAM coordinates between 2 SAAM coordinates at a given distance expressed in NM.</p> <p>Exemple: what are the coordinates between lat 2700, lon -1800 and 2700 lon 1800 and length of 1242.28 ?</p> <pre>newPointFromCoordLength.exe 1 2 3 4 5 2700 -1800 2700 1800 1242.28 <b>2946.396321 -0.013233</b> 2700 -1800 2700 1800 1242.28</pre>
<a href="#">longueur.exe</a>	<p>Used to provide the great circle distance (in NM) between 2 coordinates (in SAAM minutes decimals)</p> <p>Example: what is the length between 2700 -1800 and 2700 and 1800 ?</p> <pre>longueur.exe 1 2 3 4 2700 -1800 2700 1800 <b>2484.577326556251</b> 2700 -1800 2700 1800</pre>
<a href="#">azim.exe</a>	Used to get the azimuth (expressed in degrees) between 2 coordinates (expressed in SAAM minutes decimals)

	<p>Example: what is the azimuth between point lat 2700 lon 0 and point lat 2597 lon 1179 ?</p> <p>azim.exe 1 2 3 4 2700 0 2597 1179</p> <p><b>89.995311314327822</b> 2700 0 2597 1179</p>
qqsearch.exe	<p>Equivalent of SAAM menu Transform / matching. Allows to extract lines of a data file based on matching keys provided by a key file.</p> <p>Example: How to extract flight crossing any airspace present in traffic.t5 file (= key file), processed with traffic.so6 ?</p> <p>The key file is traffic.t5 which contains flights key in column 1 (so taken by default), the data file is traffic.so6 (flight key are in column 17) from which we want to extract lines matching with key file:</p> <p>qqsearch.exe key=17 traffic.t5 traffic.so6 &gt; traffic_match.so6</p>
qqjoin.exe	<p>Like qqsearch but merge in the resulting matched lines the key file line information (warning: if several keys appear in key file and you want to get all combinations you must use mqjoin.exe)</p> <p>Example: how to put together all information coming from so6 and t5 files ?</p> <p>The key file is traffic.t5 with key file in column 1 (so taken by default), data file is traffic.so6 (key file is in column 17, note the slightly changed command line parameters "key2" instead of "key"):</p> <p>qqjoin.exe key2=17 traffic.t5 traffic.so6 &gt; traffic.so5_t5</p>
inside.exe	<p>Used to get the information if a set of points (expressed in SAAM coordinates, in minute decimals, located using position starting at 1) are inside or outside a single 2D shape (expressed in an .are file). Output is the same set of points preceded with a flag indicating its relative position:</p> <ul style="list-style-type: none"> <li>* 3 on a shape vertex</li> <li>* 2 on a shape side</li> <li>* 1 inside the shape</li> <li>* 0 outside the shape</li> </ul> <p>Example: is that coordinates 2700 0 lying in Bordeaux ACC ?</p> <p>inside.exe BordeauxACC.are 1 2 0</p> <p>2700 0</p>

	1 2700 0
detourage.exe	

### Some examples of practical SAAM bash script commands

Alias	Description
<a href="#">animGaseCos.ksh</a>	Generate a TDV file with associated files (elementary sector file and collapse sector file) which animate the sector configuration. Takes Gase files as input. Possibly use extractGase.ksh before (see below)
<a href="#">extractGase.ksh</a>	Keep only complete set of gase information (airblock, elementary, collapse, config, opening scheme) for the list of countries given in a text file.
<a href="#">computeNetworkLoad.ksh</a>	computes the load of the input traffic file on the input network file discarding technical !.... CFMU points. The output files are a loaded network and a tab separated txt file that can be opened in a spreadsheet. First column then is the segment name, and second column is the integer load
<a href="#">countSo6.ksh</a>	gives number of flights in input traffic.so6 parameter Example: countSo6.ksh 20090627_m1.so6 <b>25708 flights 659990 in lines</b>
<a href="#">ArrDep.ksh</a>	Process sorted airport movements (arrivals + departures) from input traffic so6 or fld files (required) and provides output on console (so possibly must be redirected in an output file) with output format is: movements ICAO_code lat lon  Example: how to get all airports movements for the 27 of June 2009 ?  ArrDep.ksh 20090627_m1.so6 > airports_load_20090627.txt
<a href="#">joinX.ksh</a>	Takes in parameters all input files and join them by the first column (the key). Output line (which contains all fields for all input files) will be produced if the key is found in ALL input files. Output file is always called "joined.txt"  Example: how to put together the following files: time.txt (format is: key flight_time), length.txt (format is key flight_length) and info.txt (format is key ICAO_departure ICAO_arrival callsign aircraft) ?  joinx.ksh time.txt length.txt info.txt
<a href="#">LessThanTimeInSector.ksh</a>	Filters pair of files input.t5 and input.so6 providing new pair of t5 file and so6 file for which flights staying less than given time are kept  Example: how to get flights staying less than 1 minutes in any sectors ?  LessThanTimeInSector.ksh traffic.t5 60 traffic_less_60.t5 traffic.

	so6 traffic_less_60.so6
<a href="#">maxFL.ksh</a>	<p>Process for every flight the max Flight Level. Takes an input so6 traffic file and generates output on console (so possibly must be redirected in an output file) with output format: flight_key max_FL</p> <p>Example: how to get max FL for all flight from 27 of June 2009 ?</p> <p>maxFL.ksh 20090627_m1.so6 &gt; 20090627_m1_maxFL.txt</p>
<a href="#">So6Length.ksh</a>	<p>Process for every flight the ground route length expressed in NM. Takes an input so6 traffic file and generates output on console (so possibly must be redirected in an output file) with output format: flight_key length_in_NM</p> <p>Example: how to get the route length for all flight from 27 of June 2009 ?</p> <p>So6Length.ksh 20090627_m1.so6 &gt; 20090627_m1_length.txt</p>
<a href="#">So6Time.ksh</a>	<p>Process for every flight the flight time expressed in minute decimals. Takes an input so6 traffic file and generates output on console (so possibly must be redirected in an output file) with output format: flight_key time_in_minutes</p> <p>Example: how to get the route length for all flight from 27 of June 2009 ?</p> <p>So6Length.ksh 20090627_m1.so6 &gt; 20090627_m1_length.txt</p>
<a href="#">Opt2Refx.ksh</a>	<p>Used after flight routing optimisation, to get the reference (no route changed). The input lox file is the one coming out from optimiser, while input file so6 and t5 are the ones containing all route options. By default (if no output name are given) output files are called "Ref.so6" and "Ref.t5".</p> <p>Example:</p> <p>Opt2Refx.ksh option_after_opt.lox total.so6 total.t5</p>
<a href="#">Opt2Solx.ksh</a>	<p>Used after flight routing optimisation, to get the solution (some flight can be routed to reduce global cost). The input lox file is the one coming out from optimiser, while input file so6 and t5 are the ones containing all route options. By default (if no output name are given) output files are called "Sol.so6" and "Sol.t5". Flight IDs in output files are changed to original flight IDs allowing comparison. Only parities indicates changes: red for a re-routed flight, green for a non re-routed flight.</p> <p>Example:</p> <p>Opt2Solx.ksh option_after_opt.lox total.so6 total.t5</p>
<a href="#">AirportMerge.ksh</a>	<p>Takes CFMU (Gasel format) airport coordinates and SAAM airport file and produce and enhance SAAM airport file. No items are deleted from SAAM file (generally it contains more info than CFMU file) but only new airports are added or changed when coordinates differs.</p>

## 2.5 Analysis Menu

### 2.5.1 Queries

#### Accessed by

[Menu Bar / Analysis / Query](#)

#### Purpose / Description

Query provides the user with a functionality to search traffic and treat the search result.

The search allows user to type and combine traffic criteria or to make a selection of traffic.

The result of the search can be treated in different way:

- Display the result of the search in 3D or in 2D with different graphic features (colours, thickness, with point name, with count, by flight, by flow, etc...).
- Re-directed the search result into a (sub)traffic data file, that can be queried for a refined analysis.
- Finally, result of the search can be used as an input to set rules in SAAM.

#### Input

- [Traffic file](#) (\*.so6)
- [Traffic intersection file](#) (\*.t5) (*optional*)

#### Output

- A 2D display of traffic (flows/routes) or
- A 3D display of traffic or
- Additional \*.so6 traffic files
- Input to T5 Cleaning Rules file (TCR)

#### General Procedure

1. Ensure that the following input files are located in the working directory
  - Traffic file (\*.so6 format) (*mandatory*)
  - Traffic intersection file (\*.t5) (*optional*)
  - Optional file (\*.opt) (*optional*)
2. From the menu Bar select Analysis/ Query. The Query Dialog Box will open.
3. In the Traffic field, browse for the name of the traffic file
4. In the Search field either:
  - Type in the search criteria carefully observing the rules on [Suffix](#)
  - Select a criteria item from the [Criteria Browser](#)
5. Refine the criteria by activating various options.
6. Click the GRAPHIC button to generate the display
7. Examine the Display and continue to refine the criteria if needed or generate a sub traffic so6 file, or a set of T5 Cleaning Rules (TCR).

### 2.5.1.1 Query Dialog Box

The Query Dialog Box is used to input traffic files, criteria data and various options. This window remains on-screen (above the 2D /3D display) throughout the query operation.

**Query Dialog Box**

### Parameters:

1. **Traffic field** - Input traffic file (\*.so6 format) that SAAM requires for the query. If it is the first time you are using the traffic file, the software will first write an index file (\*.ind) onto the disk. This may take a minute.
2. **T5 field** – A traffic intersection file (\*.t5) if you want to make a query for traffic within a predetermined airspace.

Note: If the traffic file is to be automatically associated with a traffic intersection file then both files must have the same name and be located in the same directory, i.e;

**My\_name.so6 (traffic file)**

**My\_name.t5 (traffic intersection file)**

All other possible t5 files that might be linked to the so6 traffic file are selectable by the drop down menu present in the t5 field (the additional name present is the t5 file name is coming from the Airspace Traffic Intersection scenario name). For instance:

**My\_name.so6 (traffic file)** can be linked to:

**My\_nameCivil.t5 (traffic intersection file calculated with Civil airspace)**

**My\_nameMilitary.t5 (traffic intersection file calculated with Military airspace )**

**My\_nameScenarioA.t5 (traffic intersection file with Airspace Scenario A)**

If a T5 file is found or manually selected with drop down menu, Query will try to establish a link between sector names found in T5 file and sector name displayed in current TDV, only for flights of the traffic file and airspace of the airspace file (present in the current TDV) for which Airspace Traffic Intersection was made. This allow to see airspace that are crossed by flights and only these ones (1). The check box "Link to Display", when unchecked do not disable airspace display that are not crossed by flights (all airspace all always displayed).

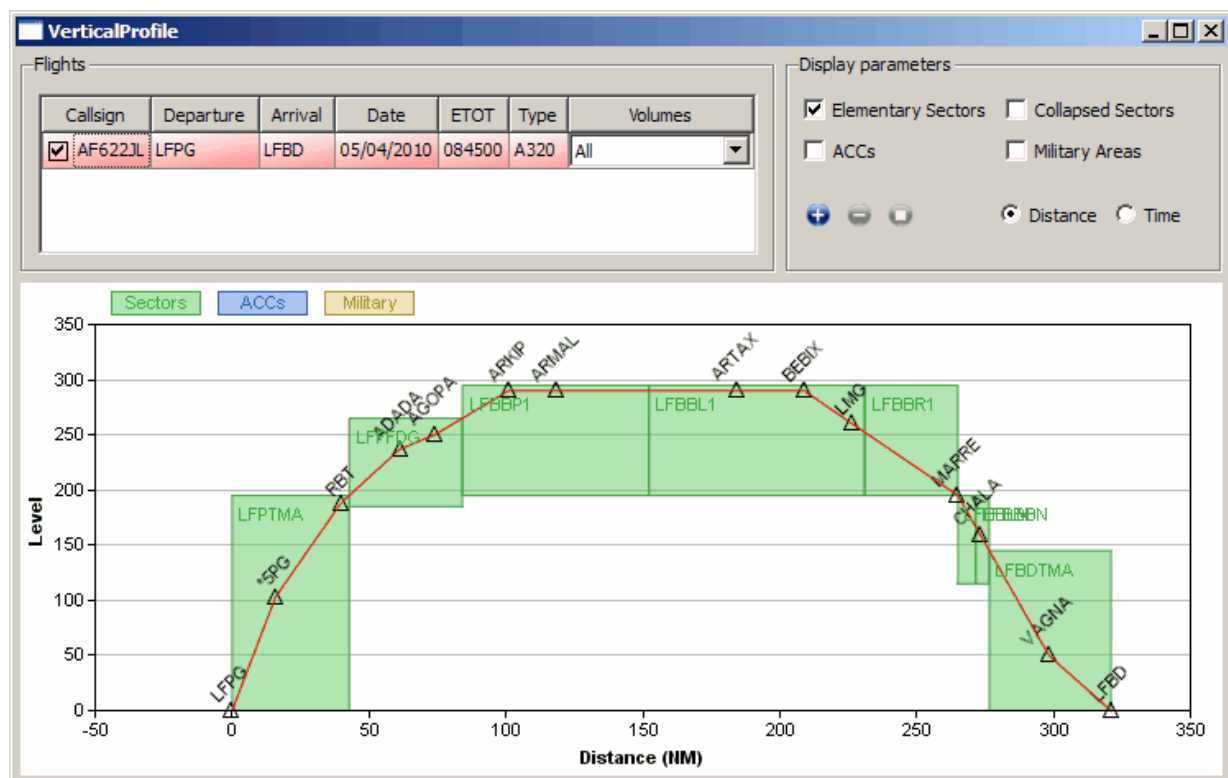
(1) if an airspace is queried -> only this airspace will be displayed (with all flights crossing it)

if flight/flight or flow/flow is checked and "link to display" is on-> all airspace crossed by flights are displayed

3. **Color button** - Opens a dialog-box that allows the choice of colours of the depicted 2D/3D traffic routing data. By default the colour is red.
4. **Transp. check-box** - When un-checked displays the depicted 2D/3D traffic routing data in solid and in transparent when it is checked.
5. **Options field** - To display routes found in a \*.lop file. This file is generated by [Route Option Generator](#) and the [Assignment](#).
6. **Edit/Create QIF file button** - Active for multi-query, Inactive (greyed) for single query. See [Multi-Criteria Query](#)
7. **Search field** - Completion of the **Search** field is mandatory and must contain valid search criteria which may be typed directly into the field or indirectly, by selecting criteria from the [Criteria Browser](#) (accessed by clicking on the **Browse** button to the right of the search field).
8. **Criteria Options check-boxes** - Allows the user to refine the appearance of the traffic display by selecting the appropriate check-boxes..
  - a. **Climb, Cruise, Desc** - Route segment display options. These options allow the user to display all or part/s of a route (e.g. if all three boxes are **checked**, then the whole route will be shown. This is the default.).
  - b. **Colour/Parity** - Deactivates the **Colour** option to show the parities of the segments. Green represents odd and blue for even. This option is also used to differentiate two different traffic files that has to be compared (see **Compare 2D traffic** (5.6.10) and **Combine So6** (5.8.14))
  - c. **Arrow** - When checked, adds an arrow to each segment showing the direction of the flight.
  - d. **Rte 2D** - Shows the route in 2 dimensions. This is necessary if the user wants to count the number of flights per ground route segment (see count below). This option can be combined with the general '**View 2D**' (option from the menu bar). Using this option while selecting only cruising or evolving traffic might lead to strange results.
  - e. **Dist.** (Distance) - Indicates length of various segments, expressed either in nm or in km
  - f. **Colour Distance** - Is used to visually depict from the selected traffic the shortest route (in green) and the longest route (in red).
  - g. **Level** - Adds a 2D or 3D label, which indicates the altitude in Flight Level at various points on the route.
  - h. **Count** - Indicates number of aircraft on segments in 2D or 3D.



- i. **Name** – Adds the name of all the segment points. Clicking on the **Name** button opens the **Label choice dialog box**. There the user can position the Label Name close to its location and/or select the type of measurement (nm/km) used in location measurements.
- j. **Vertical Profile** – To display the vertical profile of flights matching the query (limited to maximum 50 flights). A new window will automatically be opened, displaying flight information and vertical profile either by time or distance. If sectors are available and "link to display" is checked, the sectors shapes are also send to Vertical Profile (colour of sectors is not yet taken into account). Note: the minimum and maximum DFL of crossed sectors are used to provide the bottom and top level of the displayed sector shape, while the sides (left and right) boundaries of sectors shapes are based on either exact crossing time or distance.



9. **Thickness check-boxes** -Allows the user to modify the thickness of the route by adjusting the **Constant** and **Multiply Factor** values. **Constant** will add constant value to the segments and **Multiply Factor** is multiplying a value depending on the number of aircraft on the segment. A higher figure for the **Multiply Factor**, means that the difference between aircraft loads will be greatly emphasised.
  10. **STOP! button** – Stops the query.
  11. **Rules button** – Not yet implemented.
  12. **Clean T5 button** – A tool to sub-press flights in specified sectors. For more information see [Clean T5](#)
  13. **Graphic button** - Clicking on the **GRAPHIC** button initiates a graphical display of the traffic search.
- Previous routes will be cleared if the **Delete** option has been checked (located at the bottom left of the Query dialog box).
14. **SO6 FILE button** - Each traffic sample which is selected via the search criteria can be re-directed to a standard traffic file (\*.so6). For more information see [Generating a sub-SO6 file](#).
  15. **Window frame** - Sometimes it is necessary to cut off a part of the graphics in order to

obtain a better readability. This is done by defining a window:

Click on the **'Create'** button. The map will automatically change to 2d view and a tick will appear in the **'Window'** box. Define your window by dragging the mouse. By activating the **'Delete'** box, all subsequent queries will appear in the defined window only. To deactivate this function click in the ticked window box, the tick will disappear. If, the user types a name, in the field next to **Window**, before creating the window box. That window box will then be saved automatically and can be re-used later by using the **Browse** button.

16. **Manual – Flight & Flow check-boxes** – By selecting either **Flight** or **Flow** gives you the opportunity to view flights or flows one by one graphically.

After selecting one of the boxes you do the query in the usual way, you can now scroll through the data using the **PageUp/PageDown** keys. This can be used in conjunction with the **Delete** button.

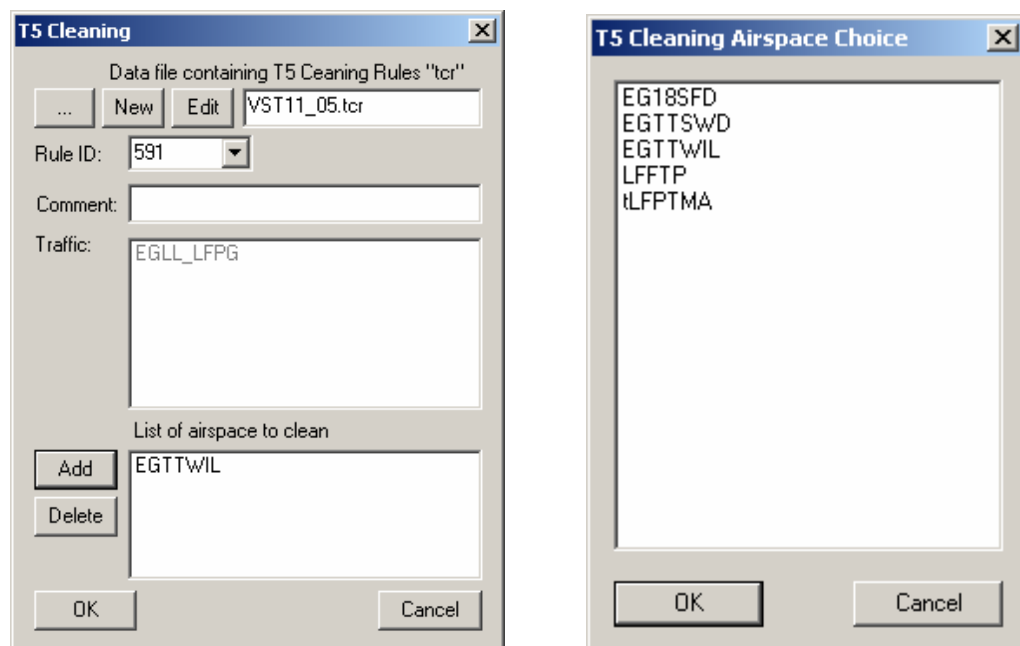
17. **validation check-box** - Not yet implemented.

18. **Delete check-box** – By default it is activated and previous queries will disappear when making a new one. When un-checked, the previous query will remain and the new query will be added to it.

19. An **Info line** - Information about the traffic after you have run your Query. For more information see [Information Line](#)

### 2.5.1.2 Clean T5

It is common that when a flight physically enters a sector, the air traffic controller in that sector doesn't take over the responsibility for it, but delegates it to another air traffic controller. This means that even though the flight physically enters the air traffic controllers sector, it doesn't add to his work load. To mimic this delegation when analysing traffic we can use **'Clean T5'**. **'Clean T5'** sub-presses flights from specified sectors, so when you do a **Sector Load** the flights are not counted in those sectors.



**T5 Cleaning dialog box**

## Parameters:

A TDV with the airspace files (\*.are and \*.sls), the same you used in [Airspace/Traffic Intersection](#) when creating the T5 file, must be in your working TDV file for '**Clean T5**' to work. To create it use [Transform/Airspace -> TDV](#).

A new dialog box is opened when you click on '**Clean T5**'.

1. **Browse (...) / New / Edit buttons and filename field** - Here you can chose between opening an existing T5 Cleaning Rules file (\*.tcr) or creating a new one. There is a button to edit the selected T5 Cleaning Rules file (\*.tcr). The explanation of T5 Cleaning Rules file (\*.tcr) format is found in your SAAM folder under **Document\FORMAT**.
2. **Rule ID drop-down list** - Shows the number of the rule you are about to add. For every rule there has to be a **comment**.
3. **Traffic field** - In the Traffic field you can see the Search Criteria you used to identify the traffic flow you are planning to sub-press. The traffic could be the result of a simple Search Criteria or the result of a more elaborate [Multi-Criteria Query](#) (QIF).
4. **'List of airspace to clean' field** - Here you can add the airspace you want to sub-press traffic from. A list with name of sectors the traffic flies through will be shown and from that list you can select one or more sectors. The **Delete** button is not yet implemented. If you make a mistake you have to use the '**Edit**' functions to correct it.

For the new added rules to have an effect, you have to run [Airspace/Traffic Intersection](#) again to create a new T5 file using the T5 Cleaning Rules file (\*.tcr).

### 2.5.1.3 Generating a sub-SO6 File

When doing a query using search criteria, a sub-So6 file is generated. This sub-file contains only flights corresponding to your search criteria. To save it, click on the '**SO6 FILE**' button. If a Traffic Intersection file (\*.t5) has been selected, an associated sub-t5 file will also be generated and saved.

The name of the sub-files (\*.so6 and \*.t5) depends on the names of the input files (\*.so6 and \*.t5) and of the search criteria. If the Traffic file (\*.so6) and the Traffic Intersection file (\*.t5) has identical name, the name of the sub-files will be '**Input\_file\_name+search\_criteria**'. There is also the possibility to give the sub-files a totally new name, by typing in the new name in the '**Save As**' dialog box that opens when clicking on the '**SO6 FILE**' button. If the input file names (\*.so6 and \*.t5) differ, there are several combinations that will have an impact on the generated names of the sub-files.

It is the location of the current TDV file that decides where the sub-files will be saved.

### Cases for sub-name generation for query

Name of S06	Name of T5	Search Criteria	User Input	Result
20MI.so6	20MI.t5	EGLL_	X	20MIEGLL_.so6 20MIEGLL_.t5
20MI.so6	20MI.t5	EGLL_	MY	MY.so6 MY.t5
20MI.so6	toto.t5	EGLL_	X	20MIEGLL_.so6 totoEGLL_.t5
20MI.so6	toto.t5	EGLL_	MY	MY.so6 MYtoto.t5
20MI.so6	20MI_S1.t5 (longer name)	EGLL_	X	20MIEGLL_.so6 20MIEGLL_S1.t5
20MI.so6	20MI_S1.t5 (longer name)	EGLL_	MY	MY.so6 MY_S1.t5
20MI.so6	20.t5 (shorter name)	EGLL_	X	20MIEGLL_.so6 20MIEGLL_.t5
20MI.so6	20.t5 (shorter name)	EGLL_	MY	MY.so6 MY.t5

## 2.5.1.4 Information Line

### Information Line

An **Info line** will appear at the bottom of the Query dialog-box with information about the traffic after you have run your Query.

If **none** of the **Flight** or **Flow** boxes are selected the Info line will contain the following information;

- 'Total'** shows the total number of flights in the traffic file (\*.so6).
- 'Select'** shows the number of flights that fulfil the search criteria.
- 'Show'** shows the number of flights displayed.
- 'Total distance'** shows the total distance of all the flights that fulfil the search criteria.
- 'Min'** shows the shortest flight route of all the flights that fulfil the search criteria. The number in brackets corresponds to the number of flights on this route.
- 'Max'** shows the longest flight route of all the flights that fulfil the search criteria. The number in brackets corresponds to the number of flights on this route.
- 'Delta'** shows the difference between the longest and the shortest flight route.

When **Flight** is selected the Info line will contain the following information.

- 'Total', 'Select'** and **'Show'** are the same as explained above.
- The following line contains flight data for the flight which is currently displayed.

0003/0403 ESSA EFVA FIN123 AT72 17h45

1 2 3 4 5 6 7

279.4nm flight ID=10497

8 9

- Number of flights which were scanned until now (including the current one)
- Total number of flights that fulfil the search criteria
- Place of departure

4. Place of destination
5. Callsign
6. Type of aircraft
7. Time of departure
8. Route length
9. Flight Identifier (SAAM unique flight identifier for a given day)

When **Flow** is selected the Info line will contain the following information.

- j. **Total**, **Select** and **Show** are the same as explained above.
- k. The following line contains flow data for the flow which is currently displayed.

0008/0403 ESSA EBBR 5000,4nm

**1 2 3 4 5**

1. Number of flights in the flow which were scanned until now (including the current ones)
  2. Total number of flights that fulfil the search criteria
  3. Place of departure
  4. Place of destination
  5. Total route length of all the flights in the flow
- l. **Min**, **Max** and **Delta** are the same as explained above.

### 2.5.1.5 Search Criteria Suffix

Certain types of search criteria require a suffix to determine the type of flight item.

The list of these characters can be obtained by leaving the **Search** field empty and click on '**Graphics**'.

The Search Criteria Suffix should be input in the query line before the data itself.

## Widening The Search

Use an asterisk (\*) to match any number of characters, or use a question mark (?) to match any single character.

Criteria	Example
All traffic	*
City-Pair	EBBR_LEMD, EG??_LFP?
Origin	LEMD_
Destination	_EDDF
Point	TGO, HAREM
Route Segment	KONAN_KOK
Callsign (suffix is ~)	~AZA123, ~BAW2*
Country-Pair	ES_LG, ED_ED
Flight ID (suffix is =)	=12345
Aircraft (suffix is :)	:A320, :B73*
Airline	~KLM*
Query in file (QIF) (suffix is either " or ')	"my_own_query

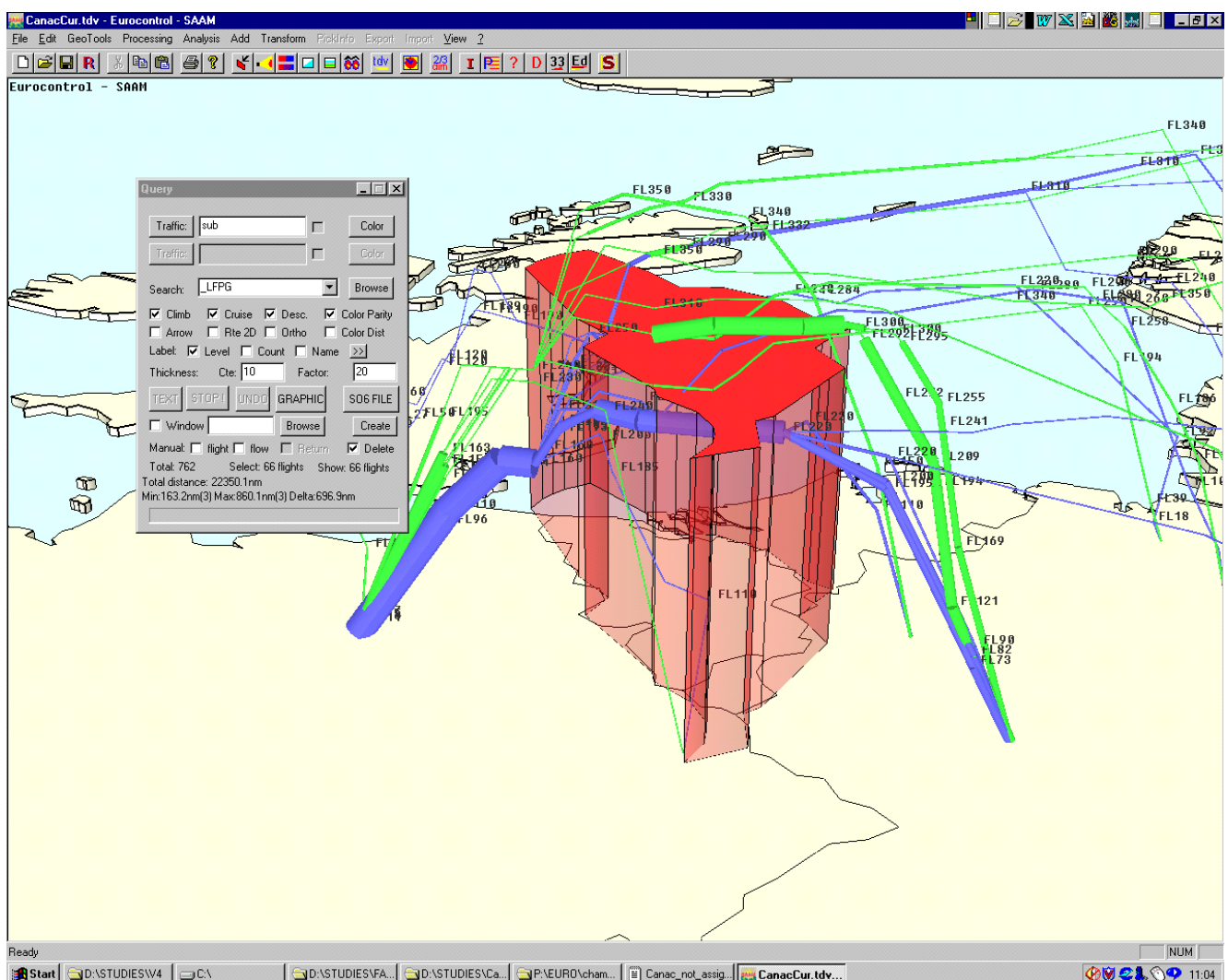
If a traffic intersection file (\*.t5) is selected you can make a query on airspace names.

Criteria	Example
Airspace (suffix is >)	>name

It's possible to select flights taking off from or landing in an airspace or both, by adding the '\_' (underscore) at the end or beginning of the of the airspace name.

### Examples:

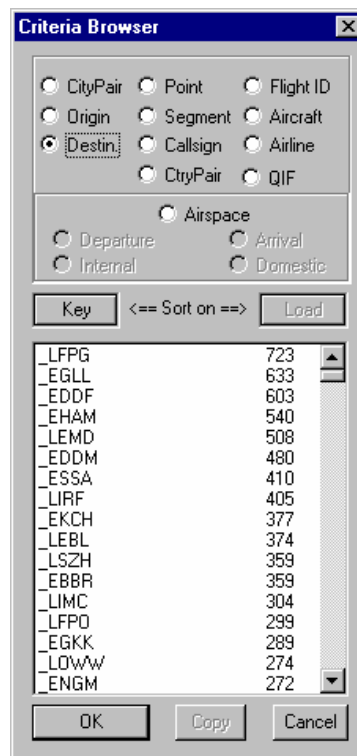
Departures from Switzerland	>LS*_
Arrivals to Switzerland	>_LS*
Domestic flights within Switzerland	>_LS*_



Example of a Query display

### 2.5.1.6 The Criteria Browser

The **Criteria browser** provides the user with an alternative or pre-defined method of completing the search field. The **Criteria Browser** window is activated by clicking on the ‘**Browse**’ button to the right of the ‘**Search**’ field in the query dialog box. The user may then chose items from a list of criteria and quickly transfer them to the ‘**Search**’ field.



**Criteria Browser dialog box**

## Parameters

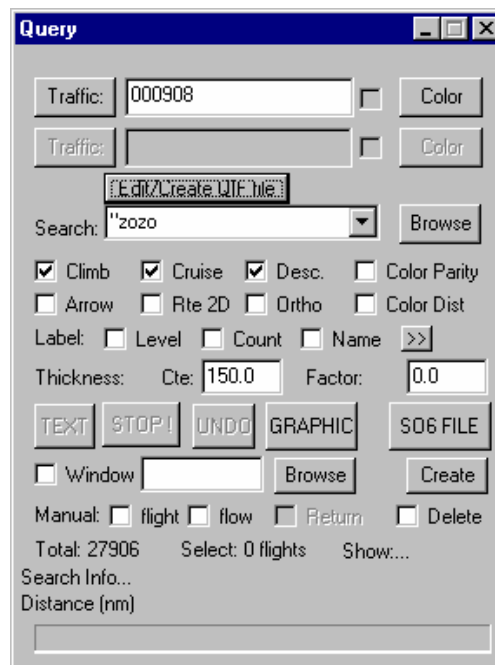
**Criteria Item Selection** – To select a criteria item the user may either;

- Double click on the criteria item, which will close the Criteria Browser and transfer the selected item to the **Search** field or
- Single click on the criteria item, which will highlight it and make the **Copy** button active. At this point, the user has the option of storing the selected criteria into the copy buffer and paste where he wants, for instance in a QIF file. Clicking '**OK**' closes the Criteria Browser and transfers the selected criteria to the '**Search**' field. Clicking '**Cancel**' leaves without transferring to the search criteria field, but copied data remains in the copy buffer.
- Sort Method** – **Key** or **Load buttons** - Sorts the order in which the criteria items will appear, which is either by its key or associated load. The user may switch between Key and Load.

### 2.5.1.7 Multi-Criteria Query or QIF

This feature allows the user to specify a multiple set of conditions.

In the '**Search**' field type the character + or " or ' immediately followed the name of your multiple-query file, the '**Edit/Create QIF**' file button will be automatically activated, allowing to edit the QIF file:



## QIF file generation & editing

By clicking then on the '**Edit/Create QIF file**' dialog-box you open a window (Notepad) where you can type in your criteria.

The rules and syntax of a \*.qif file are as follows;

- The first line is always '1' (it indicates the number of the version).
- Empty lines are ignored and spaces (or tabulation) between criteria are accepted.
- Each consecutive line indicates an additional condition (logical AND).
- The symbol '|' is used to indicate the logical 'OR' condition.
- The '!' symbol is used to indicate the logical AND NOT condition.
- The last line must always be terminated with a carriage return ( ).
- For certain search criteria you have to use a suffix , exactly like for a single search criteria (5.6.1.5)

#### Example of a QIF file

```
1
LFPG_|EGLL_
:B73?
```

The text above can thus be interpreted as :

Line 1 - Version 1

Line 2 - All flights from either Roissy departure (LFPG\_) or ( | ) Heathrow departure (EGLL\_)



Line 3 - AND that are of the Boeing 737 family (:B73?).

Every line is interpreted as a logical 'AND'. The logical 'AND NOT' is symbolised by a '!' and can be placed anywhere in front of a criteria. Thus if one wants to exclude the B738 of the selection the correct syntax should be :

**!:B738.**

Sometimes this syntax may prove a bit awkward. For instance flights that do cross sectors A or B but not A and B will be written as:

**>A | >B**

**!>A | !>B**

Flights that are neither meeting condition A nor condition B nor condition C:

**!A**

**!B**

**!C**

After the \*.qif file is completed, save it by clicking on the relevant icon and execute your query by clicking on '**GRAPHIC**' as usual.

In order to minimise the syntax errors, a copy/paste function is included. Copying is done within the Criteria Browser window. The data can be pasted into the \*.qif window.

#### **A useful QIF file**

**1**

**>ESMM3**

**!>\_ESMM3**

**!>ESMM3\_**

**!>\_ESMM3\_**

**!>ESMM3<**

The text above can thus be interpreted as :

Line 1 - Version 1

Line 2 - All flights flying through sector ESMM3. Flights that are spending a short time in the sector, and that are not taken into account because of the parameters set in Airspace Traffic Intersection process will not be taken into account here either. By default this value is 30 seconds.

Line 3 - AND that are NOT departing from inside the sector ESMM3

Line 4 - AND that are NOT arriving to an airport inside the sector ESMM3

Line 5 - AND that are NOT local flights to sector ESMM3, which are leaving and entering the sector.

Line 6 - AND that are NOT local flights strictly inside the sector ESMM3

This means, that you will only get the over-flyers of sector ESMM3. Sector ESMM3 must be defined from ground (FL 0) for it to work.

Note: each line in a qif file are limited to 10K bytes.

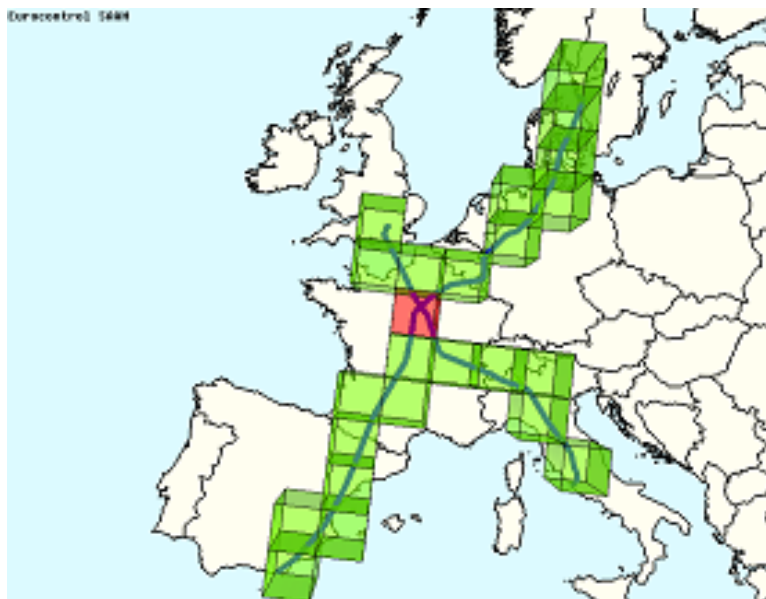
## 2.5.2 3D Densities

### Accessed by

[Menu Bar / Analysis / 3D Density](#)

### Purpose / Description

3D Density provides a method of graphically displaying the density of a wide range of traffic related data within an airspace. The airspace is sub-divided into smaller blocks called density cells and the colour of a cell within the airspace indicates its density.



The user must first define the size and position of the airspace and the density cells. The segment load figures within a certain volume are added together to produce one figure, which is the traffic load within that volume. The basic unit of a density map is a cell, which is an elementary airspace volume.

### Input

One of the following file types;

- [A traffic file \(\\*.so6\)](#) or
- A flight demand file (\*.fld) or
- A point file (\*.tdp) or
- A segment file (\*.seg) or
- A conflict file (\*.conf)

### Output

- Graphic (see below)
- Statistic (see below)
- Airspace (see below)
- Terrain (see below)
- ALL\_TO\_FILE (see below)

## General Procedure

The procedure to produce a 3D density map is carried out in three distinct steps.

### Step 1

1. Specify the size and position of the air space volume that will be used to create densities by completing the fields in the **Space Size** frame.
2. Specify the Cell size parameters by completing the fields in the **Cell Size** frame.
3. Click on **Create**. The airspace will appear on the screen map as a yellow overlay.

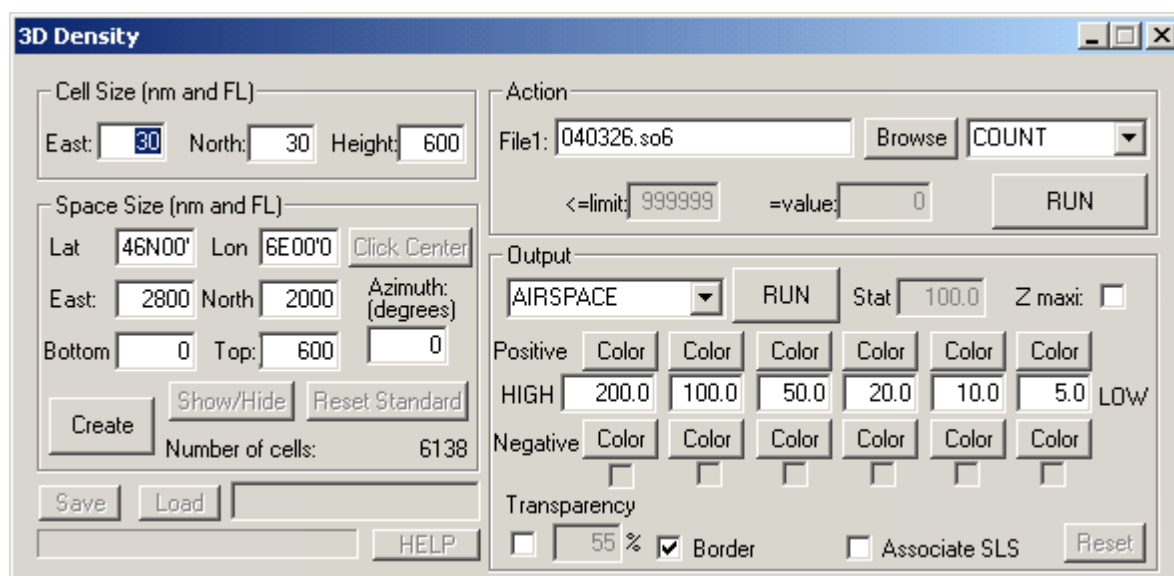
### Step 2

Complete the fields in the Action frame

4. Select the input file.
5. Determine the type of computing that will create the density cells
6. Click on **RUN**

### Step 3

7. Set up the fields in the Output frame
8. Click on the other '**RUN**' button in the Output frame in order to update the map



**3D Density Dialog Box**

## Parameters

This dialog-box is divided in 3 parts : airspace definition (Cell & Space size), cell processing (Action) and cell appearance (Output).

- 1) **Cell Size** (nm and FL) *frame*
  - a) **East field**– E/W cell size in nm
  - b) **North field** – N/W cell size in nm
- 2) **Height field** – height of cell (FL)
  - a) **Space Size frame**
    - b) **Lat field** – N/S centre of airspace in DD N/S MM.M
    - c) **Lon field** – E/W centre of airspace in DD E/W MM.M
    - d) **East field** – E/W airspace size in nm

- e) **North field** – N/S airspace size in nm
- f) **Bottom field** – Lower level of airspace
- g) **Top field** – Upper level of airspace
- h) **Azimuth field** – Airspace orientation in degrees (0 = N)
- i) **Click Center** button – Not yet implemented
- j) **Show/Hide** button – Not yet implemented
- k) **Reset Standard** button – Not implemented
- l) **Create button** – Creates the specified airspace
- m) **No of Cells info line** – Number of cells within the specified airspace
- n) **Save** button – Not yet implemented
- o) **Load** button – Not yet implemented
- p) **HELP** button – Not yet implemented

### 3) Action frame

- a) **File 1 field** – Mainly this will be either a traffic files (\*.so6) or a conflict count file (\*.conf). However, it is also possible to input a flight demand file (\*.fld) to produce traffic assessment approximations, a point file or a segment file.
- b) Drop-Down list
  - **COUNT** - adds values
  - **UN\_COUNT** -subtracts values

These two features are very useful for making a comparison between two traffic samples. By calculating comparative traffic densities it is possible to have an instant broad picture of where traffic is increasing or decreasing.

  - **CHECK** - Does not need a file. Cells having a value less or equal to the limit (specified in the '**<= limit**' field) will have their values changed to "value" (specified in the '**= value**' field). Purpose: hide cells having low values to identify/isolate cells having important values.
  - **QUICK T5**– Generates a small T5 ( called \*.QT5) based on traffic intersection with cells.
  - **RESET** - Used with a file. The trajectories of the file which are crossing cells having a value greater or equal to the limit (specified in the '**<= limit**' field) will have their values changed to 'value' (specified in the '**= value**' field). Purpose: identify/isolate the high density/dangerous areas that are crossed by a set of flights.
  - **DIVIDE\*100** – Not yet implemented

### 4) Output frame

#### a. Drop-Down list

- **AIRSPACE** - Displays the output on the current map and creates in memory an Airspace file (\*.are). The file will be saved and added to the TDV file once the user selects 'File/Save or Save as'.
- **STATISTIC** – Displays a dialog box which contains statistics concerning the loads of the cells.
- **GRAPHIC** - Displays the output on the current map and creates in memory specific TDV objects called 'cubes'. Each cube represent a cell. As usual these objects will be saved in the current TDV file once the user selects 'File/Save or Save as'.
- **TERRAIN** – Not yet implemented
- **ALL\_TO\_FILE** - Writes all cells in an \*.are file, without any display. This is very convenient when millions of cells were processed by 3D density and need to be written in a file for further processing (like complexity). Fields of \*.are contains: field "flights" the current cell value, "surface" the surface in NMxNM, "sector\_num" the volume in NMxNMxFL, "flight time" the cell value divided by surface, and "value 1" the cell value divided by the volume.

- b. Run button* – Generates the output
- c. Stat field* – Allows the user to specify the range used in the STATISTIC output.
- d. Z maxi checkbox* – When 3D cells are generated and the cell height is smaller than the height of the total airspace, then there will be more than one cell in a vertical column. Z maxi allows to select/display/save the most loaded cell per column.
- e. Positive color buttons* – Sets the colour of cells that have a value within the specified limits. When making a comparison between two traffic samples, cells with an increased value will be displayed using these colours.
- f. Negative color buttons* – Sets the colour of cells that have a decreased value when making a comparison between two traffic samples
- g. High/Low fields* – Sets the limits for different colour displays
- h. Transparency checkbox* – Displays the cells transparent
- i. Transparency % field* – Defines the level of transparency
- j. Border checkbox* – Creates a border around each cell
- k. Associate SLS checkbox* – Highlighted only when Airspace is selected in the drop-down list above. Creates an Airspace file (\*.sls), which contains the level information about each created cell.
- l. Save button* – Not yet implemented
- m. Load button* – Not yet implemented
- n. Reset button* – Not yet implemented

**N.B.:** The base unit in the case of conflict density is 0.5 as a conflict occurs between (at least) two aircraft. Thus, in order to see all aircraft, the lowest threshold should be set at 0.5.

### 2.5.3 Conflict

#### Accessed by

[Menu Bar / Analysis / Conflict](#)

#### Purpose / Description

Conflict provides an indication of conflicting traffic within a traffic sample.

#### Input

- [Traffic file \(\\*.so6\)](#) (one or several)
- Optionally, one option file (\*.lox)

#### Output

- One or several Conflict file ([\\*.conf](#)) depending on numbers of Runs and number of input Traffic files
- One or several Perturbation file (\*.prt) depending on numbers of Runs and number of input Traffic files
- Statistics about the detected conflicts in a \*.txt file.

#### General Procedure

1. Browse one or several traffic file (\*.so6)
2. Browse one option file (\*.lox) (*optional*)

3. Specify a date and time interval.
4. Specify the parameters for uncertainties.
5. Specify vertical and horizontal separation.
6. Specify the filtering parameters
7. Specify a horizontal distance in NM and step interval in seconds.
8. Specify a vertical separation. Default is RVSM. Note: if you select CUSTOM the fields "Above:" and "Below:" appears allowing user to type values for vertical separation above and below aircraft.
9. Input deviation values to Uncertainties frame. By default and usually users let the default Nb run to 1.
10. Start the run by clicking OK

**Conflict**

**Inputs**

Traffic (.so6) 20100618\_m1\_1104\_FINAL.so6

☐ Options (.lox)

**Calculation period**

Start date and time 25/01/1986 13:00:00

End date and time 28/06/2011 15:20:00

Calculation step 10 seconds

**Uncertainties**

Number of Runs	Average delay	Standard deviations
1	120 seconds	120 seconds

**Vertical separation**

☒ RVSM ☐ No RVSM ☐ Unlimited

☐ Custom: 20 FL below and 30 FL above.

**Horizontal separation**

Distance 5 NM

**Filtering**

Minimum FL 100 FL

☐ Select a single flight

**Outputs**

Scenario name NEW\_conflict\_output

Conflicts (.conf) 00618\_m1\_1104\_FINAL\_NEW\_conflict\_output\_1.conf

Perturbations (.prt) 100618\_m1\_1104\_FINAL\_NEW\_conflict\_output\_1.prt

Statistic (.txt) stats2.txt

Level of details ☒ Heavy ☐ Light

**Conflict Dialog Box**

## Parameters:

1. **Traffic field** – one or several traffic file (\*.so6)
2. **Option file field** – one option file (\*.lox)
3. **Start/End date and times dates/times** – Used to define the time frame. Caution: the input file is scan in the middle to propose a date that might be not good, check it ! This

option is invalidated is several input files are browsed. If the time or date are outside of current input traffic no conflicts will be found !

4. **Calculation step** *field* – Defines the time interval at which the calculation is done.
5. **Uncertainties** *frame* – As aircraft may not always take off on time, the user may input deviation values to take this into account. Several runs may be done with the same traffic sample: the first run uses the quoted departure time whereas subsequent runs distribute the aircraft along a Gaussian normal curve where the user can change both the average and standard deviation. Each run will generate two output files (\*\_xx.pert, \*\_xx.conf).
  - **Nb Run** – Number of runs
  - **Average** – The average delay applied to all aircraft in seconds.
  - **St Deviation** (Standard deviation) – A high percentage of all aircraft will be delayed within this time frame in seconds.
6. **Vertical separation** *Check-boxes* – Determines the vertical separation. Aircraft with less vertical separation will be considered to be in conflict.
  - **RVSM** – 1.000 ft
  - **NO RVSM** – 2.000 ft above FL290
  - **Unlimited** – All aircraft within the horizontal separation minima will be considered to be in conflict, irrespective of which flight level they are at.
  - **CUSTOM** - the vertical separation is user defined with values expressed in FL above and below the level of the aircraft.
7. **Horizontal separation** *value* – Defines the horizontal separation minima. Note: if the horizontal or the vertical separation is small, the calculation step could be reduced to increase precision (and quality).
8. **Filtering**
  - **Minimum Flight Level** *value* – Can be used to avoid conflicts close to the ground (at airports) and at take-off and landing levels.
  - **Select Aircraft** *field* – Type in flight Id to calculate conflicts only with this flight.
9. **Step** *field* – Defines the time interval at which the calculation is done
10. **Level of detail** *check-boxes* – Specifies the amount of information provided about the conflicting flights in the output conflict file.
  -
11. **Output**
  - **Scenario name** *field* – Text used to construct the name of the output file. Can be empty.
  - **Conflicts** *field, read-only* - Name of the first generated conflict file (\*.conf).
  - **Perturbations** *field, read-only* - Name of the first generated perturbation file (\*.prt).
  - **Statistic** *field* - Select the name of the file which will contains the statistics of the computation.
  - **Level of details** *check-boxes* - Select **Heavy** (Gives information about the flight Id of the aircraft in conflict, were it happens, how long it lasts, at which level etc ....) or **Light** (Will just give the flight Id of the conflicting aircraft.)

**N.B.:** Each run produces a \*.conf (conflicts) file and a perturbation file (\*.prt) The perturbation file retains the perturbation parameters, i.e. information about how much delay have been applied to each aircraft.

At present the \*.conf file can only be used to:



- Create conflict [densities](#),
- Calculate [analytical workload](#),
- Generate a TDV animating conflicts (and associated data) when conflict are calculated for ONE flight, see [transform/conflict Traffic Cut to Animation](#).

## 2.5.4 Surface / Volume

Not yet implemented

## 2.5.5 Route Length

### Accessed by

[Menu Bar](#) > [Analysis](#) > [Route Length](#)

### Purpose / Description

Route Length provides a broad analysis of the route extensions and The Route Length Extension tools aims at assessing network efficiencies.

### Input

- One or several [Traffic file/s \(\\*.so6\)](#), maximum three files

### Output

- A Text file (route\_length.txt)
- (optional) a detailed delimited text file (\_detailed.txt)

### General Procedure

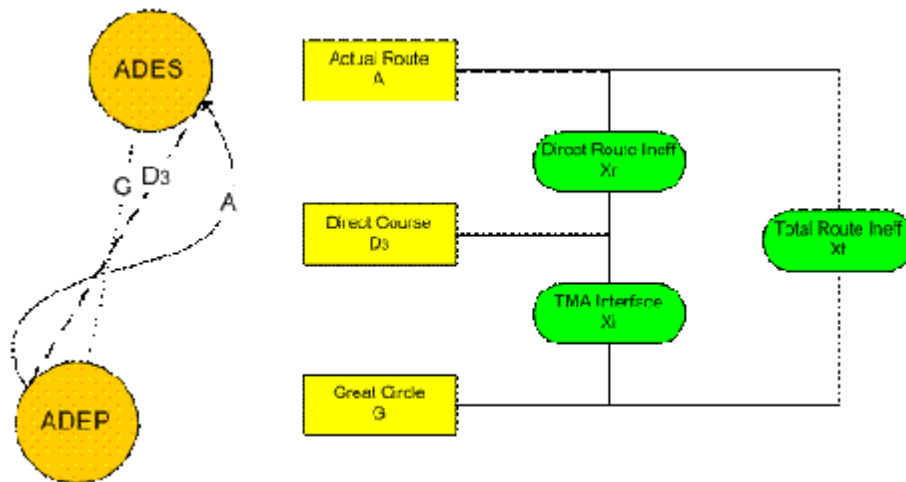
1. Browse one or several input traffic files (\*.so6)
2. Set a prefix to the output file name (optional)
3. Input SID / STAR lengths or set a TMA size
4. Set the domestic flight filter (optional)
5. Click RUN to launch processing

## Parameters

1. **Input files (So6) - file 1/file 2/file 3 fields** - Browse one or more traffic file/s (\*.so6)
2. **Output files frame** –
  - a) **Name prefix (+(-detailed)\_rte\_length.txt field** – Here you can select a prefix to the output file name. By default it is the first part of the first selected input file name.
  - b) **Output detailed comparison (M1 w.r.t. M3) check-box** – When ticked, a detailed output comparison between file2 (e.g. M3) extension and file1 (e.g. M1) extension is produced by city-pair. If used, only two input files can be selected. The comparison results are produced according to the [RNDSG methodology](#).
3. **Parameters frame** –
  - a) **SID/STAR length for orthodromic route (nm) field** - Entered as a corrective and will be added to the direct route (great distance from airport origin to airport destination)
  - b) **SID/STAR length for routes in input (nm) field** - Entered as a corrective and will be added to the flown route as described in the input traffic file/s.
  - c) **TMA size (radius in nm) field** – Creates a circle around each airport with a set radius in NM. The route length for a flight is then calculated as the length of the route between two circles, ADEP and ADEST. This way to calculate route length is based on the [RNDSG methodology](#).
  - d) **Keep only intra-european domestic flights check-box** – Route length will only be calculated on these flights.
    - **Intra-european flights**
    - **European flights + Mediterranean countries and part of the Russian federation.**

## RNDSG methodology

We are only looking at the en-route part of the route. We do not take into account TMA route part in the route length computation. To do so, we can either subtract a fixed route length for SID and STAR, or remove the route part lying in the first 30 NM around departure and arrival airports (TMA route part extraction).



A Actual Route flown by the aircraft (average route length in CFMU Model 3)

D3 Direct Course between TMA exit and entry points (according to CFMU Model 3)

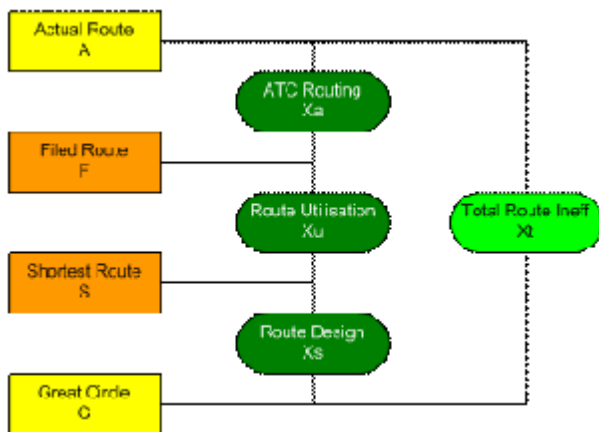
G Great Circle distance between the origin and destination TMA

Xr Direct Course Inefficiency, the relative difference between A and D3

Xi TMA Interface Inefficiency, the relative difference between D3 and G

Xt Total Route Inefficiency, the relative difference between A and G

Another method of breaking down Flight Efficiency distinguishes the causes for route extension:



A Actual route flown by the aircraft (average route length in CFMU Model 3)

F Filed Route (average route length in CFMU Model 1)

S Shortest Route available (shortest route flown in CFMU Model 1)

G Great Circle distance between the origin and destination TMA

Xa ATC Routing Inefficiency, the difference between A and F (generally negative)

Xu Route Utilisation Inefficiency (selection process), the difference between F and S

Xs Route Design Inefficiency, the difference between S and G

Xt Total Route Inefficiency, the difference between A and G

Combining these two methods provides the following matrix of inefficiencies and the method of calculation:

	Total Route Design	ATC Routing	Route Utilisation
Total Route Inefficiency	$X_t = A - G$	$X_a = A - F$	
$X_u = F - S$	$X_s = S - G$		
En-route inefficiency	$X_r = A - D_3$	$X_{ra} = X_a - X_{ia}$	
$X_{rf} = X_u - X_{iu}$	$X_{rs} = X_s - X_{is} = S - D_s$		
TMA Interface	$X_i = D_3 - G$	$X_{ia} = D_3 - D_1$	
$X_{iu} = D_1 - D_s$	$X_{is} = D_s - G$		

To make this calculation possible, 2 extra values have to be taken into account:

D1 Direct Course between TMA exit and entry points (according to CFMU Model 1)

Ds Direct Course between TMA exit and entry points (according to the shortest route available)

## 2.5.6 Maximum FL

### Accessed by

[Menu Bar / Analysis / Maximum FL](#)

### Purpose / Description

Calculates the distribution of the flights from a traffic file per Maximum FL (different from RFL).

### Input

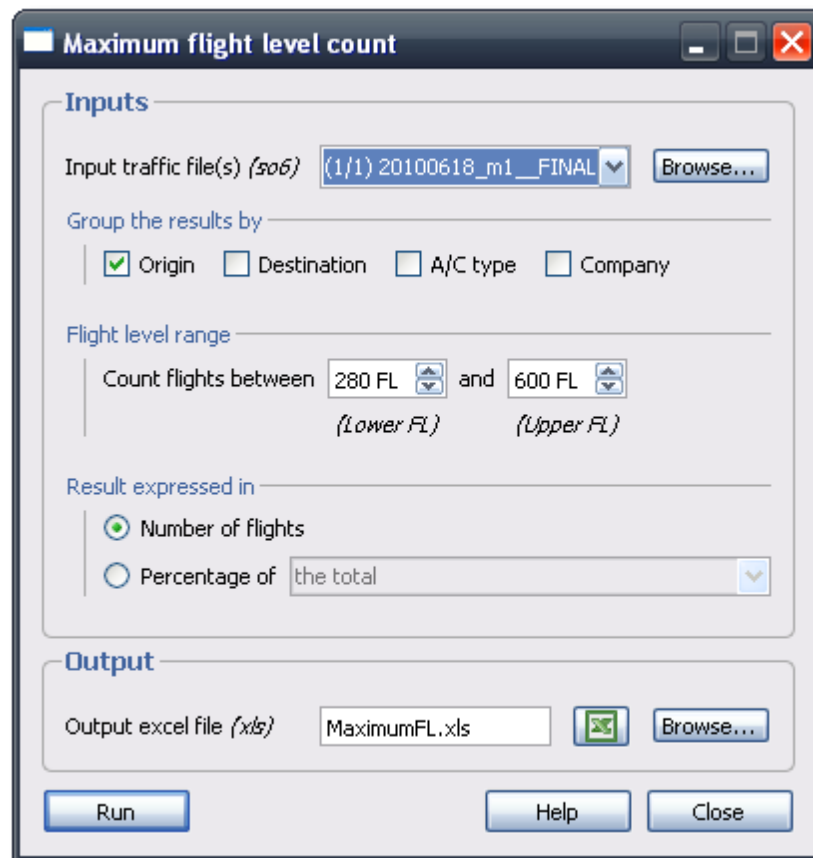
- A set of [traffic file/s \(\\*.so6\)](#)

### Output

- A text file which is automatically imported into Excel.

### General Procedure

1. Specify an input traffic file/s (\*.so6)
2. Select a criteria (*optional*)
3. Select a flight level range
4. Select the format of the result.
5. Change the default name of the result file (*optional*).
6. Click Run.



**Maximum Flight Level Traffic Count Dialog Box**

## Parameters

1. **Input Traffic field** – Browse one or more Traffic file/s (\*.so6)
2. **Group result by checkboxes** – Select the criteria to group the flight of a traffic. By default no criteria is selected, and the results will contain one line of statistic for each traffic file.
3. **Flight level range field** – Select the flight level range where the distribution of maximum FL must be computed.
4. **Result expressed in checkboxes** – Select the format of the results.

## 2.5.7 Airspace Load

### Accessed by

**Menu Bar / Analysis / Airspace Load**

### Purpose / Description

Airspace Load gives an assessment of the amount of traffic within airspace volumes (in most cases sectors).

## See Also

Traffic Intersection file (\*.t5) is dependent on [the Airspace/Traffic Intersection](#) process

## Input

- Traffic Intersection file (\*.t5)
- Sector order file (\*.ord) (*optional*)

## Output

- Airspace basic count file (\*.t5\_tdm.txt). Lists the number of flights within the selected volumes using the full traffic input.
- An Airspace load file divided in time slices file (\*.gr1) using [SAAM chart](#)

The result of the processing is a text file, which can be imported in Excel or in a text editor.

## General Procedure

1. Specify the Traffic intersection file (\*.t5)
2. Select the options required (*optional*). Default values are OK.
3. Click on the '**Run**' button to run processing.



**Airspace Load Dialog Box**

## Parameters

1. **T5 field** – Input \*.t5 file (multiple file selection is allowed)
2. **Order field** – If you don't want all sectors in your \*.t5 file to be processed, or if you want

the sectors to be in a specific order in the output, use an Order file (see [format of \\*.ord file](#)).

3. **Advanced parameters button** – Opens the [Advanced parameters](#) dialog box
4. **Scenario name field** – Add a specific suffix to the output file names initially based on the input file name.
5. **View checkbox** – Check it to automatically view the results after the processing has been launched.
6. **Run button** – Starts the Airspace Load process

### 2.5.7.1 Advanced Parameters Dialog Box

## Parameters

### Calculation of entry rate frame

#### 1. Calculation method radio buttons

- a. **Entry Rate as calculated by CFMU and NEVAC** - Another method to calculate Entry Rate. This method is set as Default. The method is explained in [Entry Rates as calculated by CFMU and NEVAC](#).



- b. **Sliding Hourly Entry Rate (SHER)** - One method to calculate number of aircraft entering a sector during one hour. It is explained in chapter [Sliding Hourly Entry Rate](#)
- 2. **Method calculation parameters frame**
  - a. **CYCLE / NON-CYCLE drop-down option list** - CYCLE is default. Traffic at the end of the day that are still in the air are re-injected in the sample as if they took off the day before at the same hour. NON-CYCLE just looks at traffic departing during that day.
  - b. **Time-slice duration drop-down option list** - is usually set to 15 minutes but can be changed according to the user's needs
  - c. **Period duration drop-down option list** - is usually set to 60 minutes but can be changed according to the user's needs.
- 3. **Additional output values checkboxes** - To add more data to the output
  - a. **Instantaneous Number of Aircraft**, which gives the number of aircraft that are observed within the sector at any minute during the interval. An aircraft will be counted during the time it actually flew in the sector. The value depends on the option selected in 'Calculation of the interval value'.
  - b. **Average Time** - Average time aircraft spent in the sector during the interval, in minutes.
  - c. **Peak of Instantaneous Aircraft Count**. If checked, the maximum instantaneous number of aircraft is output in the .tdm file
  - d. **Capacity** - Displays the Capacity figures. If checked you may select a Capacities file
- 4. **Capacity settings frame** - Will be activated when **Capacity** is checked
  - a. **Capacity file field** - Input \*.cap file
  - b. **Default Capacity field** - Capacity taken into account when it is not available in the capacity file for a given sector.
  - c. **Sort output by total capacity overload** - .When **Sort** is checked the output will be sorted by decreasing overload
- 5. **Default settings button** - Will change the parameters to default settings

### 2.5.7.2 Entry Rate as calculated by CFMU and NEVAK

The screenshot shows a dialog box titled "Calculation entry rate". It has two main sections: "Calculation Method" and "Parameters".

**Calculation Method:** Contains two radio buttons. The first, "NM Entry Rate", is selected. The second, "Sliding Hourly Entry Rate (SHER)", is unselected.

**Parameters:** Contains three dropdown menus. The first, "Cycle/Non-Cycle", is set to "CYCLE". The second, "Time-slice duration", is set to "20". The third, "Period duration", is set to "60".

Calculation of Entry Rate as done in CFMU and NEVAC, is set as default method. The principle of this method is, process the sum of entries for the n next periods making one hour (cyclic). If a flight enters a sector more than once, it is only the first entry that counts.

## Input Data

Entries	5	7	12	8	10	11	12	11	11	15	14	12	13	11	11	10	9	12	10	10	7	7
Capacity	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35

Entries - Number of entries per period of 20 minutes (default)

Capacity - Capacity (for one hour) repeated every 20 minutes

Entries	5	7	12	8	10	11	12	11	11	15	14	12	13	11	11	10	9	12	10	10	7	7
Entry Rate	24	27	30	29	33	34	34	37	40	41	39	36	34	32	30	31	31	32	27	24	19	19
Capacity	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35
Overload	0	0	0	0	0	0	0	2	5	6	4	1	0	0	0	0	0	0	0	0	0	0

Process the sum of entries for the n next periods making one hour (cyclic).

Example:  $24 = 5 + 7 + 12$

$27 = 7 + 12 + 8$

etc....

### 2.5.7.3 Sliding Hourly Entry Rate (SHER)

**Calculation entry rate**

Calculation Method

☐ NM Entry Rate

☒ Sliding Hourly Entry Rate (SHER)

Parameters

Cycle/Non-Cycle: CYCLE

Time-slice duration: 20

Period duration: 60

Calculation of the interval value: At interval bounds

Entry time centering: 0

Calculation of **Sliding Hourly Entry Rate (SHER)**, is the number of aircraft entering a sector within one hour. This figure is calculated every minute.

When calculating SHER it is assumed that an entering aircraft always stays one hour in a sector, independent of how long it actually stays. This means an aircraft will be counted for one hour.

If a flight enters a sector more than once, it is only the first entry that counts.

### Calculation of the interval value

SHER is calculated for one hour, every minute. When displaying the results of the

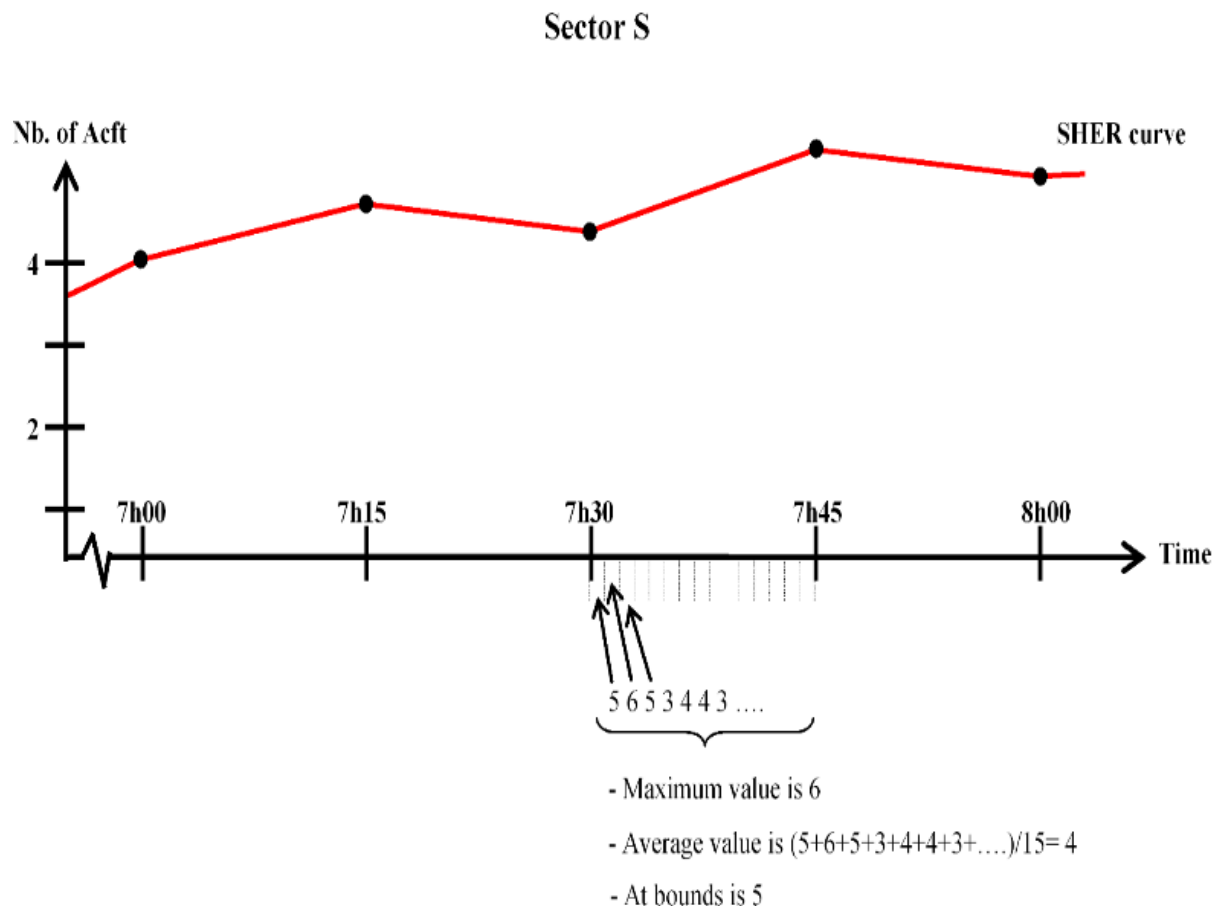
calculations as a curve in a graph you can select an interval in minutes. All the calculations within that interval will be taken into account when a value is displayed. The value will be shown over the intervals start time.

The value displayed depends on the option selected in the parameter dialog box for “Calculation of the interval value”:

- Max – The calculation with maximum aircraft entering
- Average – An average of all the calculations made during the interval.
- At bounds – The actual value of the calculation at the beginning of the interval

**Example:**

The graph below shows a SHER curve with 15 minutes interval.



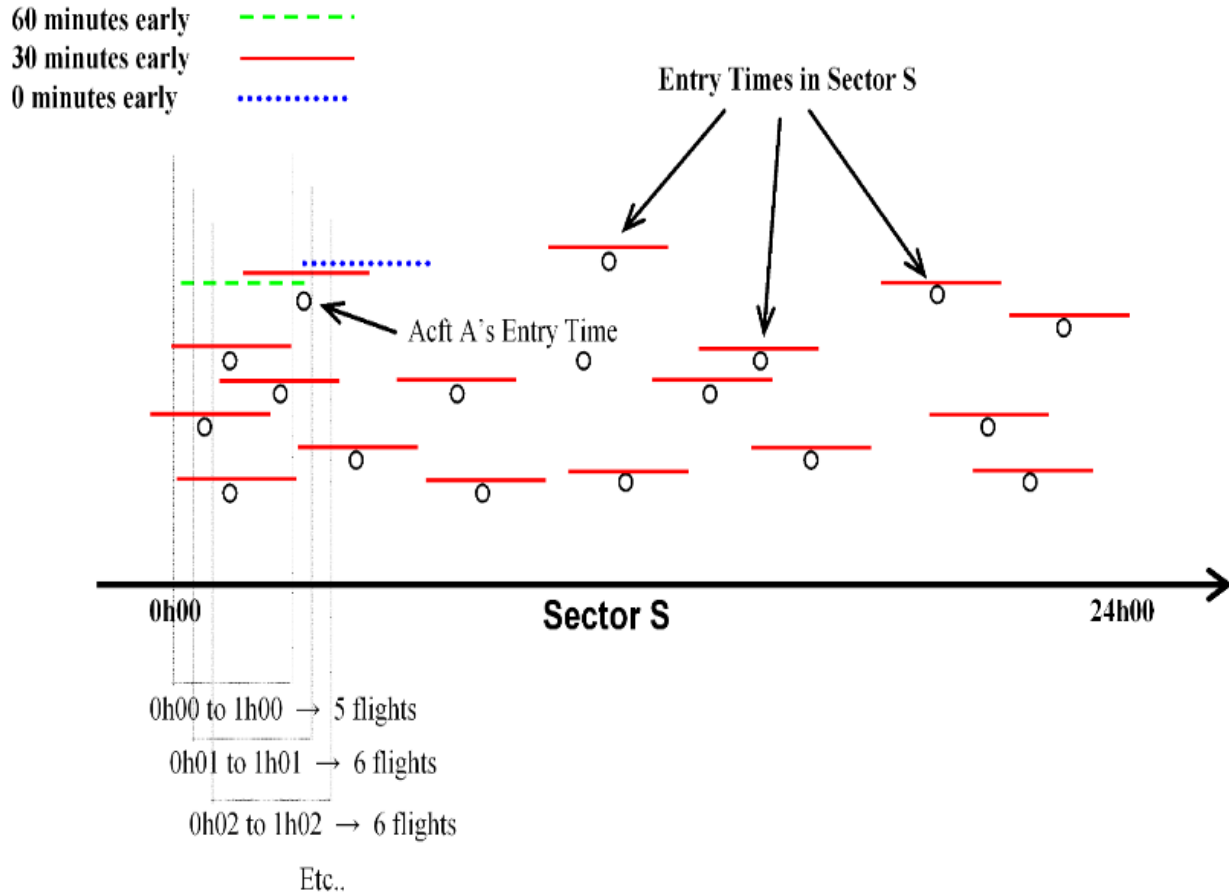
## Entry Time centering

An aircraft's entry to a sector can be viewed as a one hour long time bar, showing the time frame the aircraft will be counted. The time bars beginning and end time related to the actual entry time can vary. This means that an aircraft can be counted before it physically enters a sector, since its expected entry can add up to the load in the sector. In the dialog box there are three options of when to start counting an aircraft (“Entry time centering” drop-down list):

- 60 minutes early - The aircraft will start being counted 60 minutes before its entry time.
- 30 minutes early - The aircraft will start being counted 30 minutes before its entry time and stop 30 minutes after.
- 0 minutes early - The aircraft will start being counted at its entry time

**Example:**

In the picture below you can see that Aircraft A's entry will be counted differently depending on the option you select in the dialog box. If '60 minutes early' is selected, Acft. A will be counted between '0h00 to 1h00', which isn't the case if '0 minutes early' is selected.

**2.5.7.4 Overload**

To see how the sectors entry rate correlates with its corresponding Hourly Capacity figure, both values can be displayed at the same time in a graph. If there are more aircraft entering during an interval than the capacity allows there is an overload. Note that when using "Generic entry rate" with "Period duration" different from one hour, the hourly capacity is converted to the time basis of the entry rate in order to compute overload.

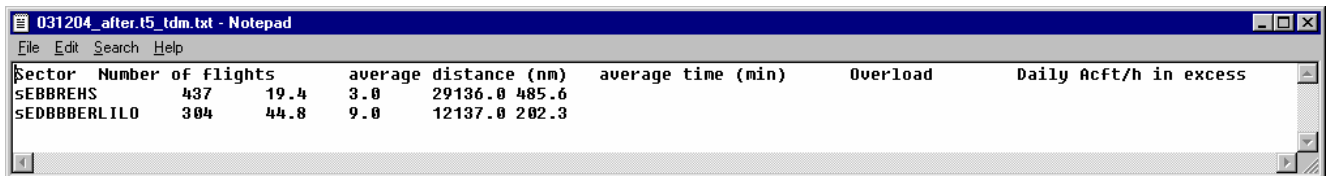
The sectors **Total Overload** is calculated by adding up all the extra aircraft entering every minute. From the sum one can't tell whether there were many aircraft during a short time period, a peak, or if there were few aircraft during a long period. 10 extra aircraft during one minute gives the same sum as one aircraft extra for 10 minutes.

By dividing the sectors **Total Overload** by 60 you will get 'Daily Aircraft/hour in excess' which appear in the .tdm file.

### 2.5.7.5 Viewing the Results

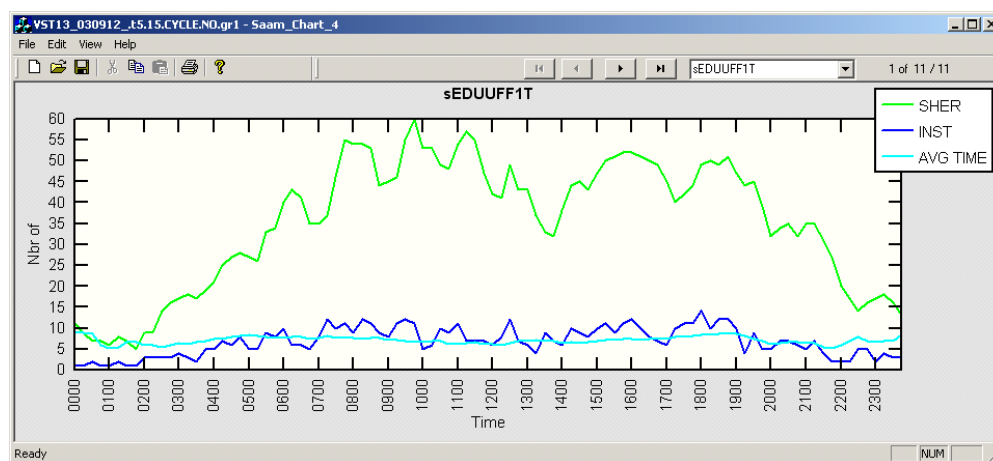
The Airspace Load process generates a \*.t5\_tdm.txt file and a graphics file (\*.gr1). If you have checked the “View output file” checkbox, these file automatically open after you have clicked ‘**Run**’. The text file (\*.t5\_tdm.txt) is an overview of the results and the graphic file will display information and curves depending on the selection made in the Parameters dialog box.

For more information about the graph go to [SAAM Graph Module](#)



Sector	Number of Flights	average distance (nm)	average time (min)	Overload	Daily Acft/h in excess
SEBBREHS	437	19.4	3.0	29136.0	485.6
SEDBBERLIL0	304	44.8	9.0	12137.0	202.3

*Example Text file*



*Example Graph file*

## 2.5.8 Combine Airspace Curves

### Accessed by

[Menu Bar / Analysis /Combine Airspace Curves](#)

### Purpose / Description

Produces a comparative Sector Load graph by comparing two input graph files

### See Also

[Airspace Load](#) function to provide input files (\*.gr1)

## Input

- Graph file 1 (\*.gr1)
- Graph file 2 (\*.gr1)
- Sector Capacity file (*Optional*)

## Output

- Graph file (\*.gr1)

## General Procedure

1. Specify input files, (Graph file 1 and Graph file 2)
2. Specify Sector Capacity file (*optional*)
3. Select Combination method
4. Specify name of output file
5. Set Sort method
6. Set filter limits (*optional*)
7. Click OK to run

**Combine Airspace Curves Dialog Box**

## Parameters

1. **Input File 1/File 2 fields** – Used to select the two graph files (\*.gr1) for combination. These files must have the same time step, number of airspace and the 3 curves SHER, Instant Count and Average Time. In **Title** you can specify the name that will be given to the curve in the graph.
2. **Output File field** – Name of output file (\*.gr1)
3. **Curve drop-down list** – Will list all the common curves found in the input files.

4. **Operation drop-down list** – Will list the different options to display the common curves.
- **NO COMBINATION: Display Curves** – Will display all common curves.
  - **Curves are subtracted** - The value of curve 1 is subtracted from the value of curve 2.
  - **% variation between curves** - : The resulting curve depicts the % of variation between curve 2 and curve 1 which is the base.
  - **Curves are added** - The value of curve 1 is added to the value of curve 2.

**Capacity frame** – There are two ways to display capacity. Either a default value or by browsing a capacity file. In **Title** you can specify the name that will be given to the curve in the graph.

When using a capacity file there might be values missing and replaced by ‘dummy’ values of either ‘0’ or ‘999’. As these extreme values will produce chaotic graphs, the user has the option to translate these values into more reasonable figures. This is done by replacing ‘0’ and ‘999’ with a specified value.

5. **Sort frame** –
- a) **Sort by drop-down list** – Lists options of different sort criteria
    - **No sort**
    - **Max Delta Value**
    - **Delta Surface**
    - **Delta square surface**
  - b) **Over Capacity by**
  - c) **Filter Sort Values**

## 2.5.9 Compare Traffic Routings

### Accessed by

[Menu Bar / Analysis / Compare Traffic Routings](#)

### Purpose / Description

Compares traffic routings processed from two input traffic file (\*.so6). This routine will compare the routings flight per flight based on flight ID (it is the reason why the two input traffic files should have the same flight). It will provide screen information on traffic routing difference for the flights that are common, number of additional flights and number of suppressed flights.

### Input

- Two Traffic files (\*.so6)

### Output

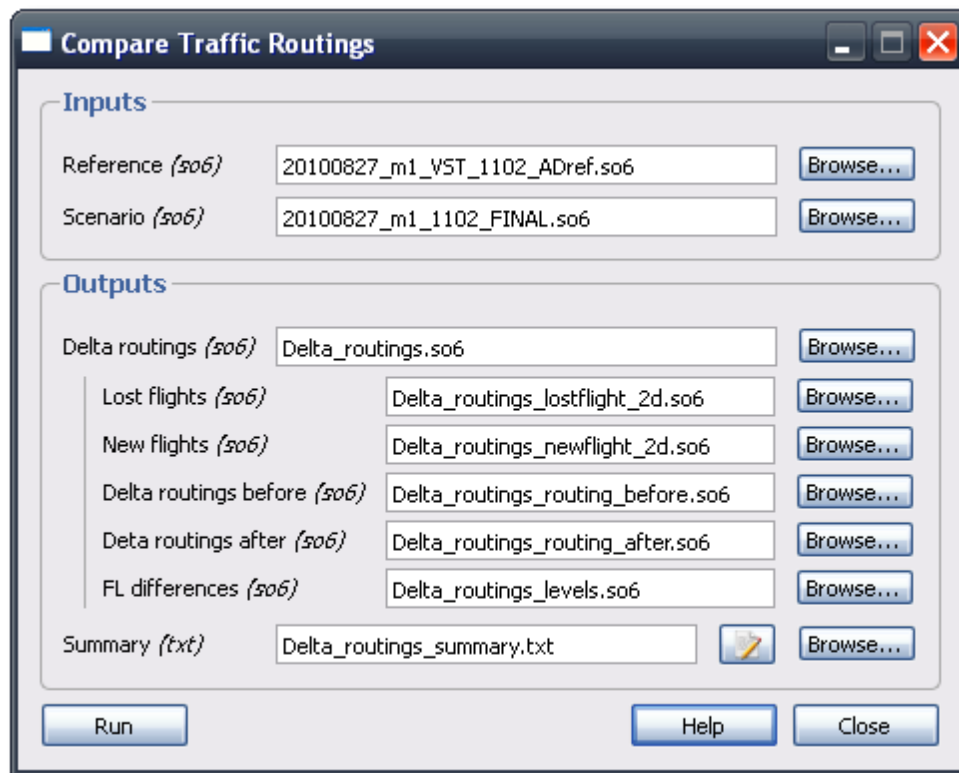
The following traffic files (\*.so6) are generated in the current working directory:

- lost flights
- new flights
- delta routing before: for the common flights, contains the reference routings
- delta routings after: for the common flights, contains the scenario routings
- delta routings: combination of two previous files, were parity was changed, to facilitate visual comparison from the Query
- delta level: only for the common flights, having the same routings, this file shows

Flight Level differences.

## General Procedure

1. Specify reference traffic file
2. Specify scenario traffic file
3. Change the output traffic file names if necessary.
4. Click 'RUN'



**Compare Traffic Routings Dialog Box**

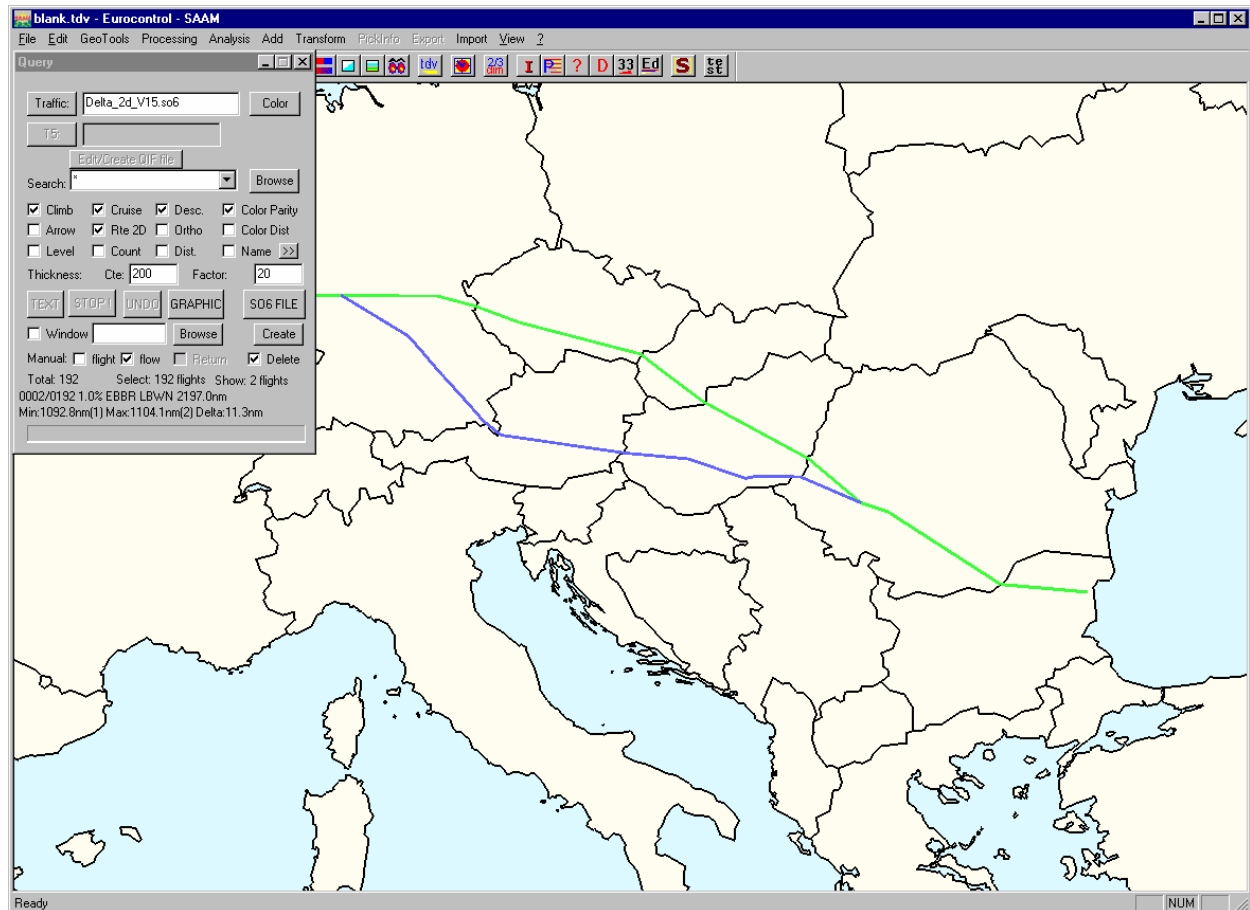
## Parameters

1. **Reference field** – Browse the reference traffic file (\*.so6).
2. **Scenario field** – Browse the scenario traffic file (\*.so6).
3. **Delta routings field** – The main output file (\*.so6). Changing the name of this file, will automatically change the name of all the sub files.
4. **Summary field** – A file which contains a summary of the comparison (\*.txt).

## Analysis of the result

You can use the [Query](#) to view the result. In the [query dialog box](#), browse the *Delta routings* file and select the options *Rte 2D*, *Colour Parity* and *Flow*. The reference traffic will be displayed in green, the scenario in blue.





*Example comparison analysis*

## 2.5.10 Airport Arrival/Departure Time

### Accessed by

[Menu Bar / Analysis /Airport Arr/Dep Time](#)

### Purpose / Description

Generates Excel charts depicting the Departure/Arrival times for Airports.

### Input

- A traffic file (\*.so6).

### Output

- An Excel file which contains the departures and/or arrivals times for the selected airports.

### General Procedure

1. Specify the **traffic** file (\*.so6).



2. Select either '**Arrival**', '**Departure**' or both.
3. Type the **ICAO code** of an airport or select "all airports".
4. Select the **date** contained in the traffic file to process.
5. Specify the **Duration**
6. Specify the **Shift**
7. Change the name of the output file if necessary.
8. Click on the **Run** button.

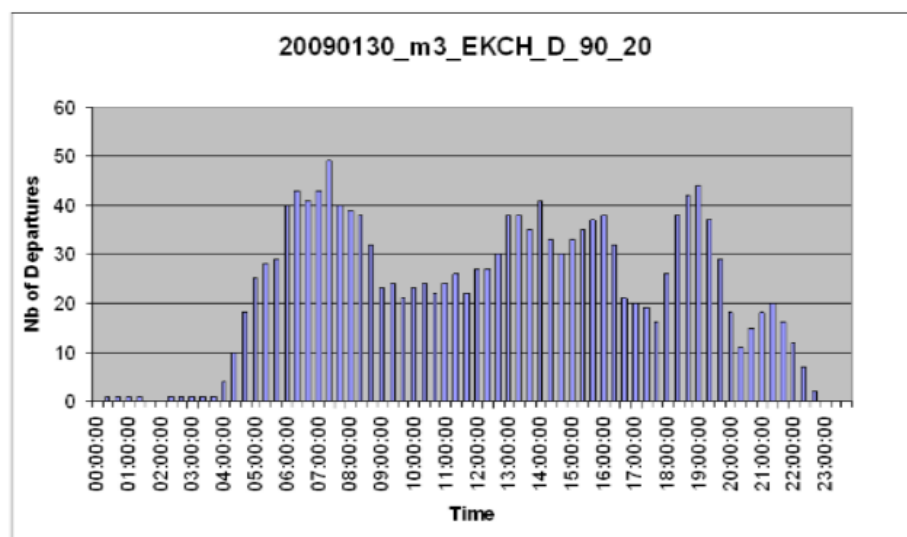


**Airport Arrival/Departure Time Dialog Box**

## Parameters

1. **Traffic field** - Browse a traffic file (\*.so6)
1. **Process all airports within the input traffic file check box** – If checked, all the airports within the provided traffic file will be processed.
2. **Arrival/Departure drop-down options** – Count the DEPARTURES, the ARRIVALS or both (ARRIVALS + DEPARTURES).
3. **For...**
  - a. **the airport with ICAO code check box** – If checked, only the airport with the specified ICAO code will be processed.
  - b. **all airports within the input traffic file check box** – If checked, all the airports found in the traffic are processed.
4. **And...**

- a. **for the date** *check box* – If checked, only the flight departing or arriving at the specified date are processed. The available list of date is automatically calculated from the traffic file.
- b. **merge the results of all date** *together check box* – If checked, no date filtering is performed. The results for all dates found in the traffic file are merged together.
- 5. **Duration field** - Specifies the time-frame which is used for the calculation of the data. For instance, when Duration is set to 90 minutes, the result for 08:00:00 will add up all the departures for the next 90 minutes, to 09:30:00.
- 6. **Shift field** - Indicates the 'step' used for calculation. Thus 20 means that results are calculated every 20 minutes.
- 7. **Output drop-down list** - Specifies the name of the output Excel file. The  button allow opening the output file in Excel. From there you can create a chart to display the result.
- 8. **Automatically open the result in Excel** *check box* - If checked, Excel will be automatically opened at the end of the process.
- 2. **Process all airports within the input traffic file** *check box* – If checked, all the airports within the provided traffic file will be processed.
- 3. **Select airport with ICAO code** *check box* – If checked, only the airport with the specified ICAO code will be processed.
- 4. **Arrival/Departure drop-down options** – Count the DEPARTURES, the ARRIVALS or both (ARRIVALS + DEPARTURES).
- 5. **Duration field** - Specifies the time-frame which is used for the calculation of the data. For instance, when Duration is set to 90 minutes, the result for 08:00:00 will add up all the departures for the next 90 minutes, to 09:30:00.
- 6. **Shift field** - Indicates the 'step' used for calculation. Thus 20 means that results are calculated every 20 minutes.
- 7. **Output drop-down list** - Specifies the name of the output Excel file. The  button allow opening the output file in Excel. From there you can create a chart to display the result.
- 8. **Automatically open the result in Excel** *check box* - If checked, Excel will be automatically opened at the end of the process.



*EKCH Departures - Graph generated by Excel*

## 2.5.11 Compare Network

### Accessed by

[Menu Bar / Analysis / Compare Network](#)

### Purpose / Description

Compares two network files (\*.ase) by segment or segment load and generates a differences network file. To compare by segment load the two networks has to be loaded, number of flights on each route segment has to be known. This can be done either by an [Assignment](#) or by transforming a traffic file into a network, [so6 to network](#).

**Note:** The readability of the result depends on the degree of difference between the input files, too many differences will produce a very complex file.

### See Also

[Add Network](#) function

### Input

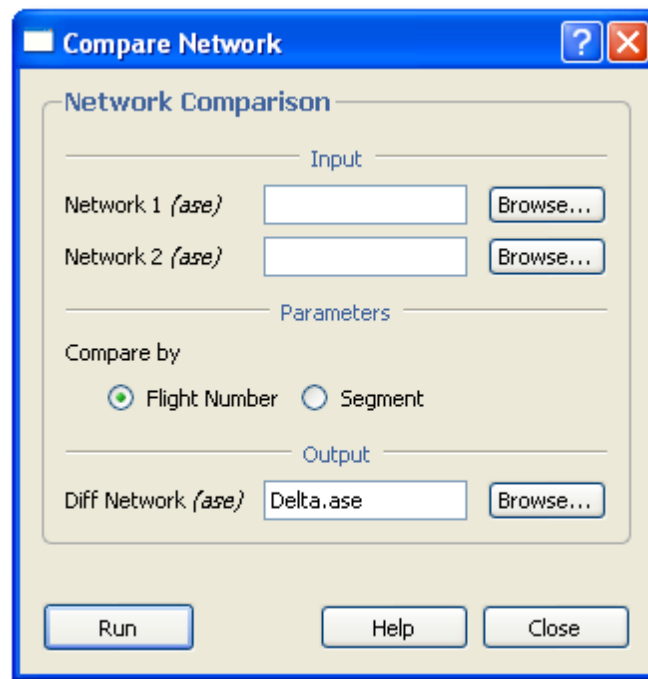
- Two Network files (\*.ase)

### Output

- Difference Network file (\*.ase)

### General Procedure

1. Specify the 1st network (\*.ase)
2. Specify the 2nd network (\*.ase)
3. Specify a name for the result file (\*.ase)
4. Select the comparison method (Flight number / Segment)
5. Click on **Run** to run



**Compare Network Dialog Box**

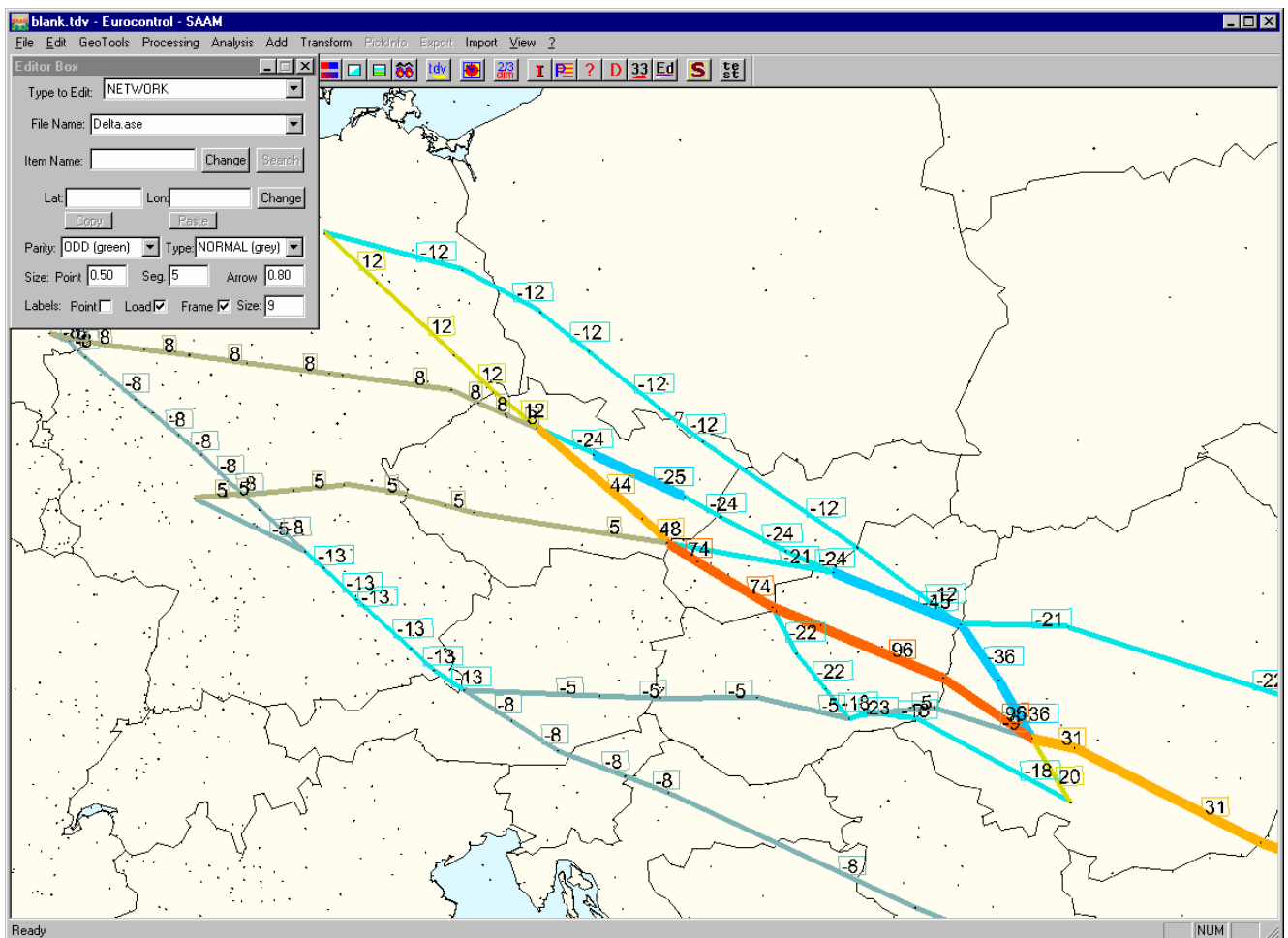
### Parameters:

- Segments that are the same in both input files, Network 1 and Network 2, are not generated in the output file, Diff Network, which means they won't be displayed.
- Once the process is completed, the resulting network can be visualized in SAAM by using the [Add Network](#) function.

### 2.5.11.1 Segment Load Comparison

1. Open [Add Network](#) dialog box
2. Click on '**Browse**' in order to load the resulting network.
3. Change the threshold to get a satisfactory display. Differences in load are usually between 100 and 0.
4. Adjustments in the width of the segments and display of the figures are usually necessary . This is done via the [Network Editor](#)

A good display would be like this :



Example of good display

One can also simply compare two unloaded networks : the display will show where segments have either been removed or added. In this case all the '**Add Network**' thresholds **must** be set to 1.

## 2.5.12 Compare Airspace

### Accessed by

**Menu Bar / Analysis / Compare Airspace**

### Purpose / Description

Compares two Traffic Intersection files (\*.t5). When using the same traffic file (\*.so6) but have differences either in the route network or in the airspace design the two scenarios can be compared using this tool.

### Input

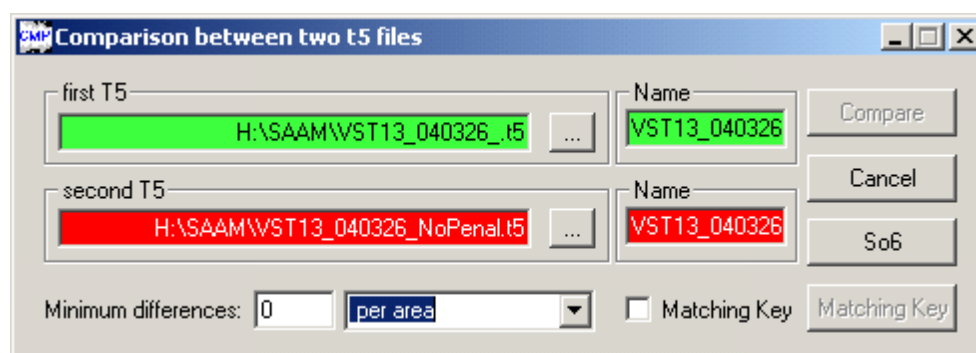
- Two Traffic Intersection files (\*.t5)

### Output

- An output text file

### General Procedure

1. Browse 1st Traffic Intersection file (\*.t5)
2. Browse 2nd Traffic Intersection files (\*.t5)
3. Specify a name for the output file (\*.txt). Default is 'Check\_Sect.txt' *optional*
4. Set the parameters *optional*
5. Click on **Compare** to run



**Comparison between two t5 Dialog Box**

### Parameters

1. **first T5/second T5 fields** – The two Traffic Intersection files (\*.t5) must be generated by using the same traffic file (\*.so6)

2. **So6 button** – To select the corresponding traffic files (\*.so6) to the t5
3. **Minimum difference field** – By typing in a number greater than 0 you can choose to see just sectors where the difference in flights is bigger than the number typed.
4. **per area / per flight drop-down list** – An option to see the differences per area or per flight.
  - **Per area** – When selected will show the number of flights going through an area.
  - **Per flight** – When selected will show the number of sectors a flight will fly through.
5. **Matching Key check box** – Activates the button '**Matching Key**'. If the airspace design is different between the two compared scenarios and you are using different sector names, by using '**Matching Key**' you can still compare these by using a key file. The key file matches one sector name to another. Create a file by clicking on '**Open**' or browse an existing file. The file format is: *sector\_name\_1 separated by a space sector\_name\_2*, this means that sector *sector\_name\_1* will be compared with sector *sector\_name\_2*.

**Example:**    **sEHAMTMAold sEHAMTMAnew**  
                   **sEBKTMold sEBKTMnew**

## 2.5.13 Route Charge

### Accessed by

[Menu Bar / Analysis /Route Charge](#)

### Purpose / Description

Route Charge is a specific module that has been developed for the CRCO. It calculates for each flight the cost due to each country. Additionally it can combine a traffic file (\*.so6) and an associated intersection file (\*.t5) to produce a route charge data file (\*.crco).

### Input

- One traffic file (\*.so6)
- One associated Intersection file (\*.t5). Note: this T5 can be calculated with the [Airspace/Traffic Intersection](#) module using ALL Route Charge Airspace Areas (generally CRCOChargeAreas.are and CRCOChargeAreas.sls). These two files are provided using menu [Import/Get Basic Airspace Data/Route Charge Set](#). It is up to you to check its content and possibly update it.
- The CRCO data: aircraft weights, unit rates and member states. These data can be provided by SAAM.

*Note:* The Route Charge Set files which is copied in your working directory when you imported then from menu Import/Get Basic Airspace were adapted to provide reasonable behaviour concerning route cost. In particular, extra route charge zones were artificially added, with fake unit rate, simply to avoid the effect of choosing a route which is cheapest because there is no route charge cost associated to a part of it. All of these files can be changed. It is up to you to check its content and possibly update it.



## Output

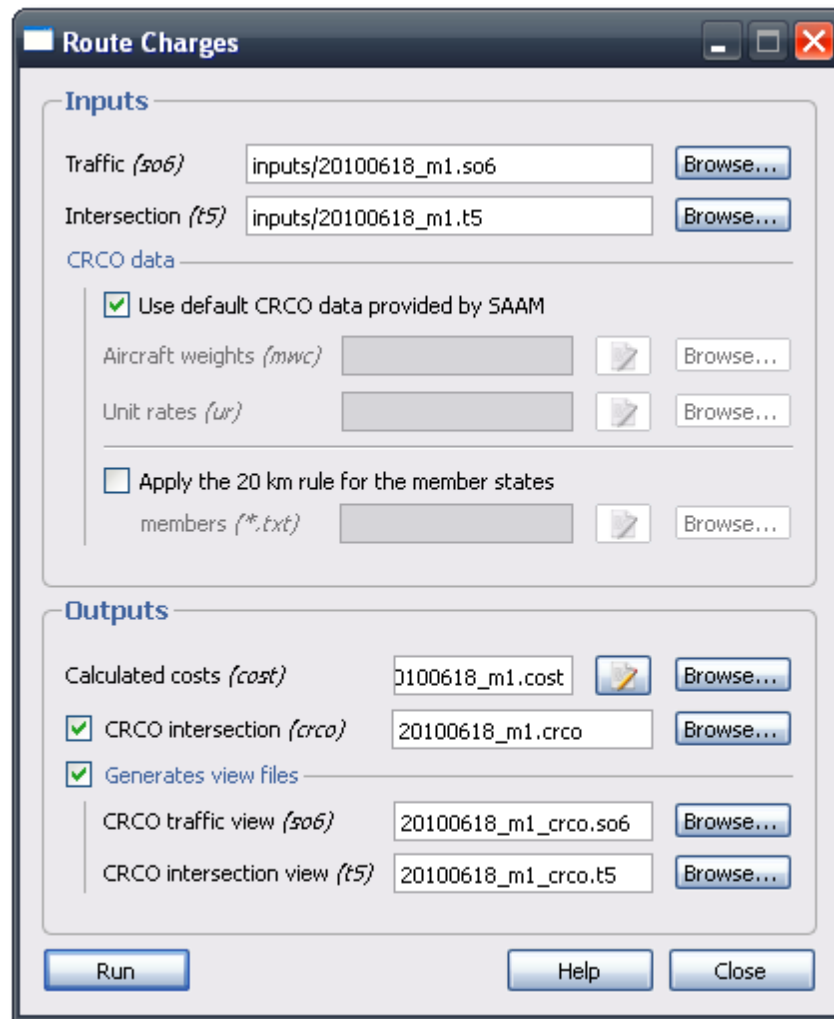
- One [cost file](#) (\*.cost) containing flight ID, crossed country, associated cost in Euro. Below you will find an extract of .cost file with same example as above (flight crossing EG and LF) with the cost expressed in EURO underlined.

```
129633734 EG 1421.77  
129633734 LF 763.53
```

- Optionally, one [CRCO Route Charge data file](#) (\*.crco). This file is more reserved to technical usage, internal to CRCO, it contains more information for each portion of flight crossing each country. For information, column 9, contains the length in Km of each part of direct CRCO distance above each countries. This length might be reduced, due to [20Km ajustement](#).
- Optionally, two additional files for [visualisation](#) (\*\_crco.t5 and \*\_crco.so6) .

## General Procedure

1. Specify the input traffic file (\*.so6)
2. Specify the associated Intersection file (\*.t5) if it is not automatically selected.
3. Specify your own CRCO data if you don't want to use the ones provided by SAAM.
4. Check "Apply the 20km rule" if needed.
5. Select the optional output you want to generate: CRCO intersection (\*.crco) or view files.
6. Click on Run



**Route Charge (CRCO Distance) Dialog Box**

## Parameters

1. **Traffic field** – Browse a traffic file (\*.so6)
2. **Intersection field** – Browse a intersection file (\*.t5).
3. **CRCO data**
  - a. **Use default CRCO data provided by SAAM** check-box – if checked, the CRCO data provided internally with your SAAM release will be used automatically. Else, if you want to modify or see CRCO data files, you can retrieve them from SAAM internal storage with the menu [Import/Get Basic Airspace Data/Route Charge Set](#), all CRCO data files will then be copied in your working directory, and you will need to browse them (see below)
  - b. **Aircraft weight field** – Possible only if field "Use default CRCO data provided by SAAM" is unchecked, Browse for your own aircraft weight file (\*.mwc).
  - c. **Unit rate field** – Possible only if field "Use default CRCO data provided by SAAM" is unchecked, Browse for your own unit rate file (\*.ur).
  - d. **Apply the 20 km rule check-box** – if checked the [CRCO 20 km rule](#) is applied to the cost calculation.
  - e. **Members field** – Browse for your own member file (\*.txt). It is the list of states

which apply the 20km rule.

4. **Calculated costs field** – Browse the name and location of the output cost file (\*.cost).
5. **CRCO intersection field** – Browse the name and location of the output CRCO file (\*.crco). Note: the generation of this file is not enabled by default.
6. **Generate view files fields** – If checked, browse additional files, a \*\_crco.so6 file and an associated \*\_crco.t5 file used for [visualisation](#). They are not generated by default.
7. **Adjust 20km check box option** - The 20 km CRCO rule is applied when calculating the \*.crco distance file. This rule does not affect the optional traffic and intersection files. Default is 'off'.

## The CRCO 20km Rule

For all flights departing or landing within airspace belonging to a CRCO member state and which departing/landing date are greater or equal to the date the member state joined CRCO, there will be a deduction in direct route length by 20 kilometres. This is done to adjust the route length not to include length for SID and STAR. The list of member states and the date they joined CRCO are specified in the file 'members.txt'.

*Important note:* For specific case of airports located close to countries boundaries, it might happened that the deduction of 20Km is propagated to possibly neighbouring different countries, as CRCO currently does it ! For instance: departure from Airport XXXX goes through country A for a direct distance of 5Km, then through country B for a direct distance of 12Km then finally through country C for a direct distance of 200Km. After deduction of 20Km the direct distance will be: 0 for country A, 0 for country B and 197Km for country C. This effect imply to calculate Route Charge with ALL countries, avoiding annoying transfer of 20Km to other, mostly wrong, recipients !

## Visualisation the results

The route charge module can generate special outputs called view files, called by default "\*\_crco.so6" and "\*\_crco.t5". These files are convenient to check graphically the direct distance from entry to exit point for each country. If, in [query](#) you activate the display of segment distance in KM, you will get same distance as indicated in column 9 of .CRCO file, except for case of 20Km deduction (see image below).



In blue the CRCO direct distance (with distance label), in red the flight plan traffic

## 2.5.14 Sector Workload (Macroscopic formula)

### Accessed by

[Menu Bar / Analysis /Sector Workload](#)

### Purpose / Description

Calculates an estimate of the workload in a sector and provides a display of the workload curve.

### Input


- A traffic file (\*.so6)
- An associated intersection file (\*.t5)

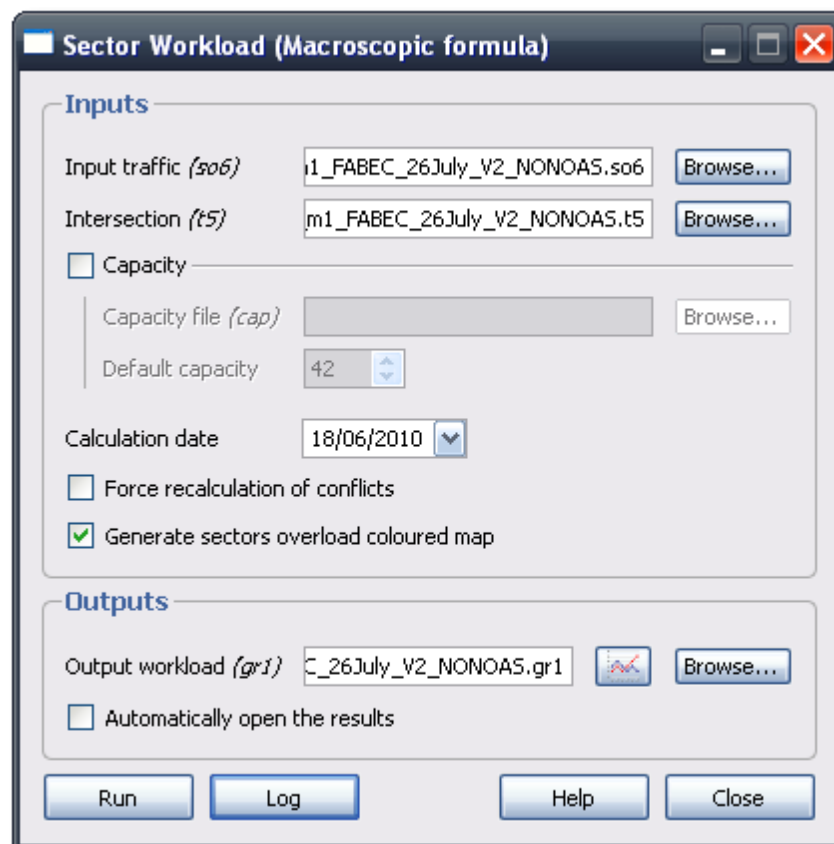
### Output

- Workload file (\*.gr1 format)

### General Procedure

1. Specify the input traffic file
2. Specify the associated intersection file
3. Select a capacity file and change the default capacity (*optional*).

4. Change the calculation date if necessary
5. Change the output Workload file if necessary
6. Click on Calculate to Run
7. Display the result graphic by clicking on  (optional).




**Sector Workload (Workload Calculator) Dialog Box**

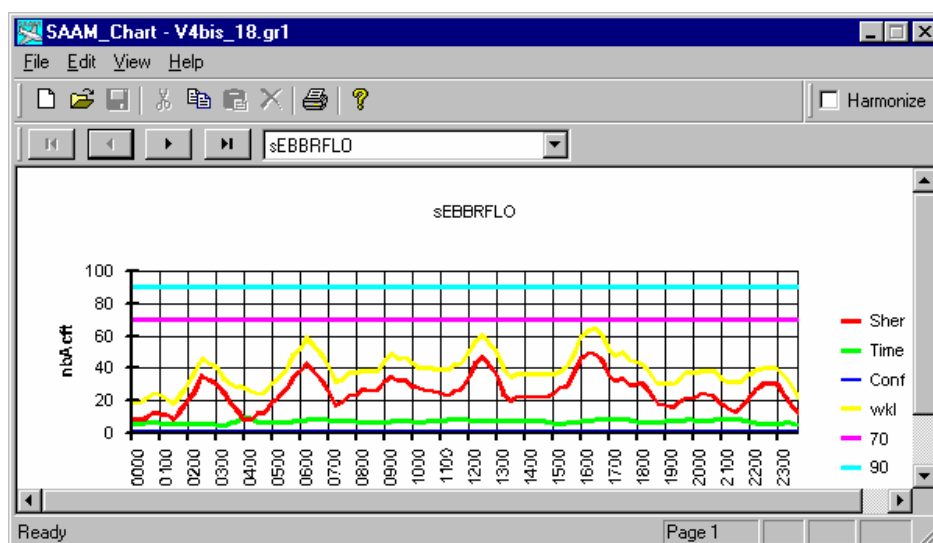
## Parameters

1. **Input traffic field** – Browse a traffic file (\*.so6)
2. **Intersection field** – Browse a Traffic Intersection file (\*.t5)
3. **Capacity check box** – Check it to activate the capacity
  - i. **Capacity file field** - Browse a capacity file (\*.cap).
  - ii. **Default capacity number** - Select to the default capacity to use.
4. **Calculation date** calendar - Select the date of calculation, useful if the traffic file has more than one day of traffic. By default it is set to the first date found in the traffic file.
5. **Force calculation of conflicts check box** – If checked the conflicts are re-calculated even if it has been previously calculated for the selected traffic.
6. **Generate Sectors Overload Coloured Map check box** – If checked, a coloured map of sector workload is automatically generated. This map will show sectors with capacity problems. It might be useful to cut the display at a given level with F9 (the map displayed in 2D has a stack of 3D sectors). A legend is automatically inserted on the map. Note: it requires to launch before ["menu/transform/Airspace -> TDV"](#) with sectors which corresponds to your analysis.

This function is accessible only if capacity is set in the dialog box. It also requires you to run "Transform / Airspace to TDV" beforehand, with the same sectors as used in the Workload

process.

7. **Output workload field** – Select the name of the output graphic (\*.gr1). Clicking on the  shortcut button allow displaying the results via the graph display module. For more information go to [SAAM Graph Module](#).
8. **Log button** – Show the content of the log file (warnings, errors, ...).



**Sector Workload - Graph Display**

## 2.5.15 Sector Workload / Capacity (CAPAN like)

### 2.5.15.1 Introduction to CAPAN like

**CAPAN-Like** is a new method of Workload calculation which has been developed for the **SAAM** tool.

This method is derived from the **CAPAN** and **RAMS** methodology and is ATC Task oriented. For each flight crossing a sector, a set of basic ATC Tasks is recorded, according to the flight profiles, the critical events of the flight through the Airspace and the conflicts detected. Each task is defined by a Position responsible (Executive / Planning Controllers / or both) and an execution time for the execution of the Task.

This method doesn't have the pretension to replace the actual Fast Time Simulation Tools, but only to improve the assessment of the simulated organisations. Fast Time Simulations and Real Time Simulations are still useful to fine tune the diagnostics.

### 2.5.15.2 Description of Input / Output Data

#### Description of Input Data

CAPAN-Like can be used directly from a SAAM scenario and requires that several files be present prior to the execution. The files listed below are compulsory, as they define the environment to be analysed (Traffic and Airspace).

- so6 - 4D Flight Profiles File containing the analysed Traffic Sample
- t5 - Intersection file which contains the Sector Penetrations of the So6 Traffic in the

Airspace Considered.

- conf - Conflict file which contains the conflicts detected in the So6 file
- sls – Sector file describing the airspace organisation
- are – airblocks used to define the sectors

The Files listed below are mandatory and describe the parameters to be used for the analysis. If they are not present, they are created automatically with predefined values, prior to the execution of the CAPAN–Like Module. To create these files for possible update, you need to type a name with correct extension in the corresponding browse dialog box, a message will appear asking if you want to create a new file, clic YES and the file will be created with standard value that you can edit and modify.

- ATC-Tasks - (file \*.clt) Contains basic parameters and the description of ATC Tasks for several class of sectors.
- Sectors – (file \*.cls) List of sectors and basic parameters (at least the class task to be linked with) to be used for each sector considered.
- General-Parameters – Contains basic parameters governing the execution of the module.

## Description of Output Data

CAPAN Like will generate several output files which can then be used SAAM or from Excel to analyse the main results of the Study. The output files are stored in a folder named ORG\_RESULTS\_CAPAN. ORG is a name given to the run via the dialog box. The output files are listed below:

- ORG\_Simlog.spt
- ORG\_Sector\_Name.xls (One file for each sector)
- ORG\_Main\_results.xls
- ORG\_Summary\_Tasks.xls
- ORG\_Table\_A4.xls
- ORG\_Graphics.gr1
- ORG\_TaskFlight.spt

Simlog.spt - Contains a detailed printout of the simulation events, with the so6-t5 and conf information for each flight. The level of details is governed by a parameter specified in general parameters parameters and the description of ATC Tasks for several sector families.

Main\_results.xls – This file contains a summary of the main results obtained par sector, like total number of flights, total Flight Times , EC and PC Peak Hour workloads, total working times, and estimated sector capacity ...

Sector\_Name.xls – For each sector simulated, an Excel file is created with a detailed distribution of the sector activity during the period considered, like the hourly traffic entry rates, the corresponding EC and PC workloads, the occupancy rates This output can be used to make more in-depth analysis of the sector activity.

TaskFlight.spt – Contain a detail description of all the ATC Tasks recorded per Flight. It can

be used for debugging purposes. This file is generated only when the debug parameter is set to a value higher than 10.

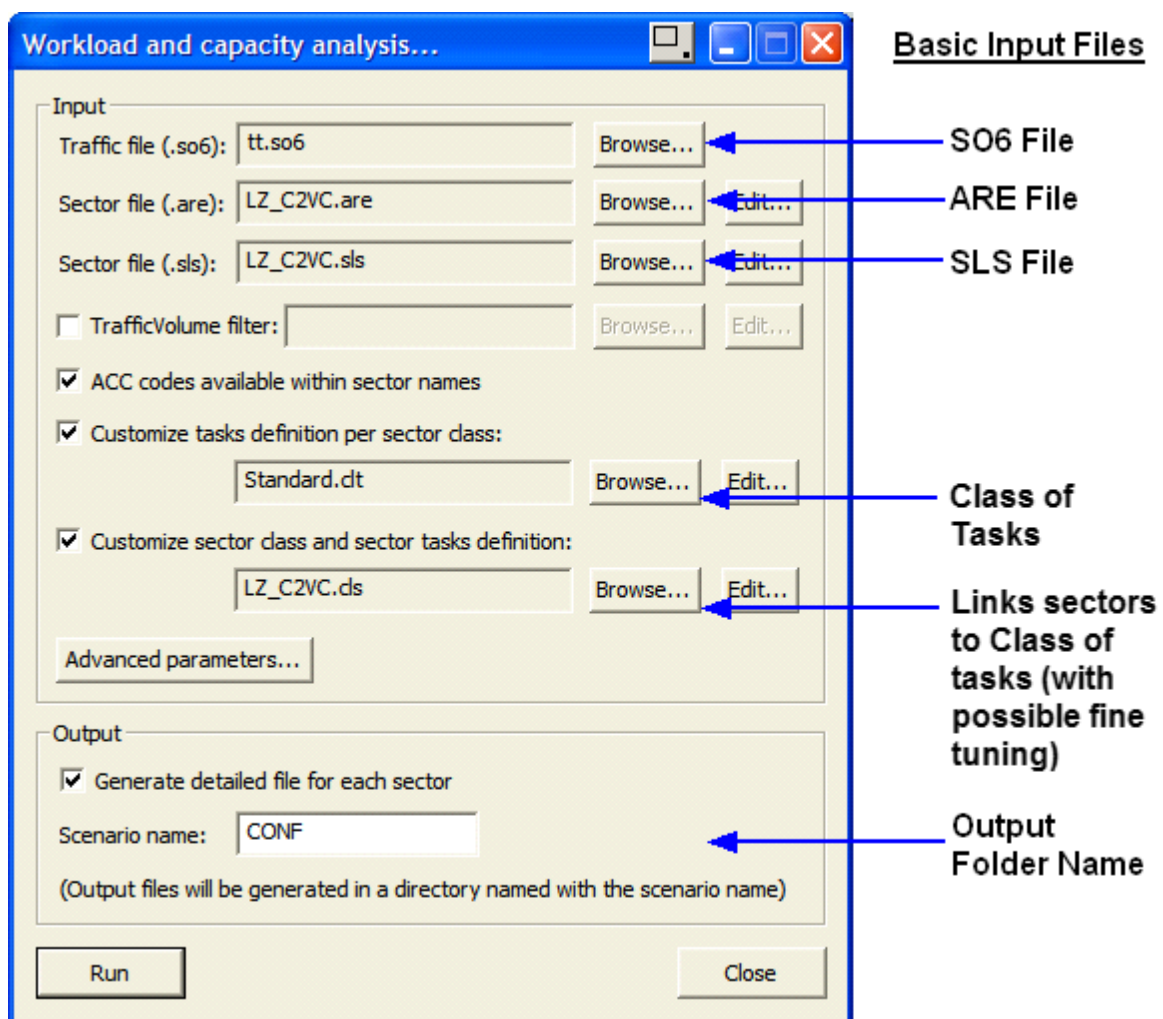
Table\_A4.xls Contains for each sector, a summary of the Total working times per category of ATC Tasks.

### 2.5.15.3 Running CAPAN like Module

The CAPAN-Like module is executed from SAAM.

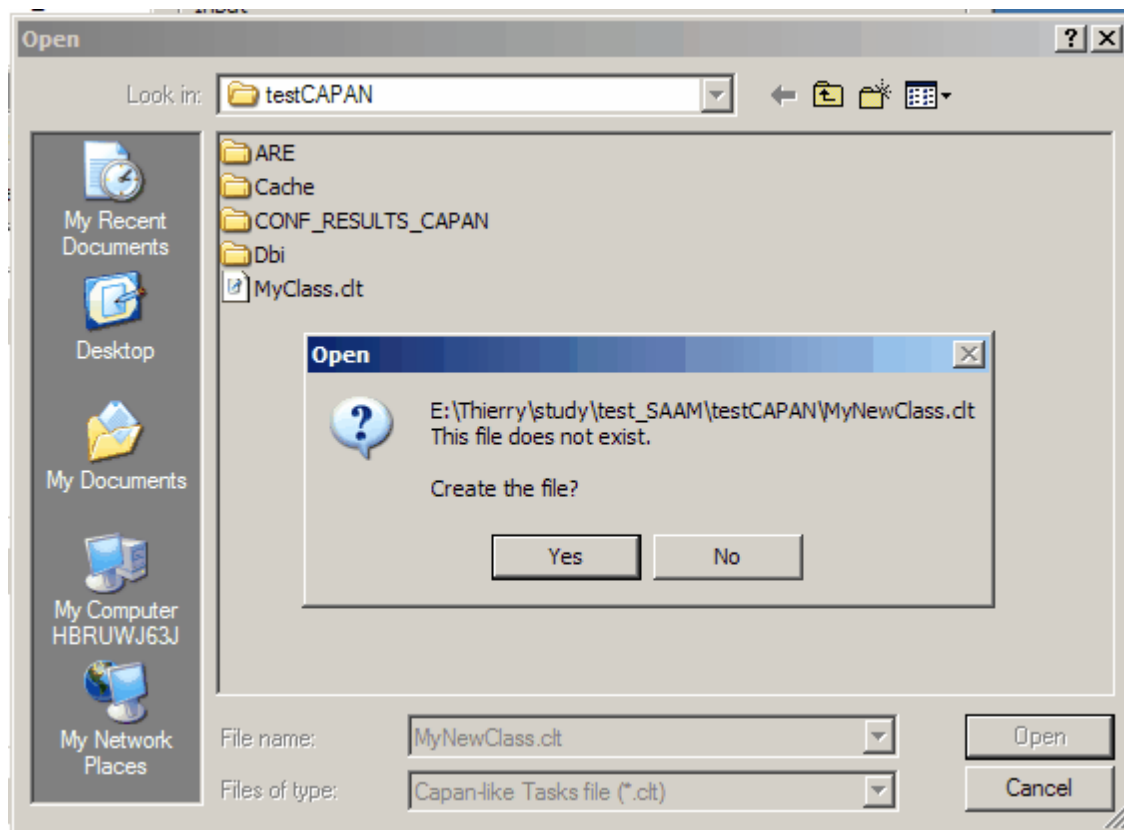
#### Menu Analysis / Sector Workload / Capacity (CAPAN like)

#### Main Dialog box

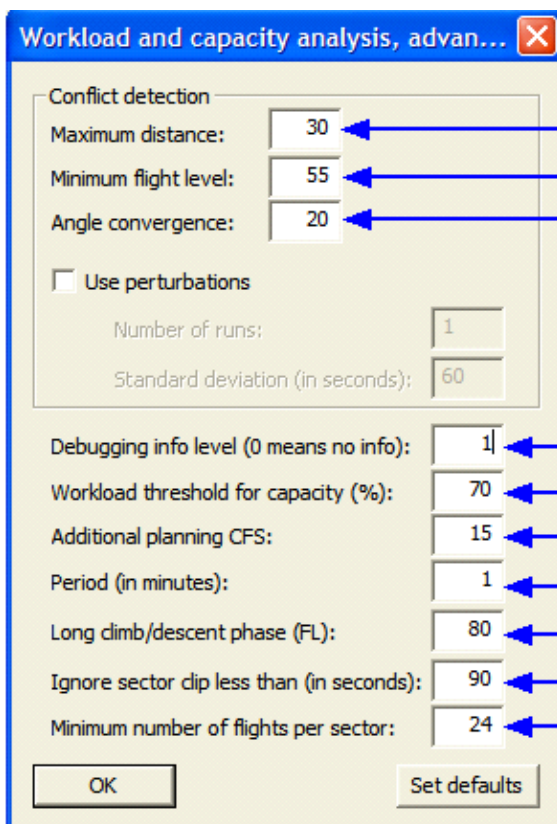


To create a new default \*.clt file and/or new \*.cls file, clic on corresponding browse button, type the name of the file with its correct extension, click on "Open" button", then a Message box will appear asking if you really want to create a new file, click "Yes" and the file will be created, its content will be automatically initialized with default values (see below):





## Advanced Parameters



### Basic Input Files

Minimum Distance for Conflict Detection (Nautical Miles)

Minimum Flight Level for Conflict Detection

Angle to use for classification of conflicts

Debugging Level

Overload Threshold (Percentage)

When more than 15 Flights in the sector, increase Conflict Search Tasks

Sliding Window for graphics (Minutes)

For long Climb/Descent phase, record intermediate messages (Feet\* 100)

If a non simulated area is found between two sectors for a duration less than 90 seconds, the gap is ignored

Ignore sector for analysis if less than maximum number of flights in the day

## DESCRIPTION OF SECTOR PARAMETERS

### Supervision thresholds

The **Supervision** threshold is a distance between two aircraft below which the controller is supposed to closely monitor the potential conflict between the two flights. The threshold can be defined for each of the nine types of conflict and is expressed in  $\text{NMS} * 100$ . A typical value of 20 NMS is generally used but for large sectors it is recommended to use higher values like 30 NMS to take account of the sector (zooming factor). Also, for along track conflicts, smaller values like 16 NMS are generally used to take account of the catching speed of the two aircraft.

### Intervention thresholds

The Intervention Threshold is the distance below which the controller is supposed to intervene in order to solve the conflict (vectoring, temporary level change, speed control). Like the Supervision Thresholds a value is defined for each conflict type expressed in  $\text{NMS} * 100$ . Generally, the Intervention Threshold is defined as one half of the Supervision Threshold.

### Maximum number of Interventions/Supervisions

SAAM only detects the conflicts and does not try to solve them. Also, depending on the origin of the so6 file used for the study, it is possible that flights are presented to a sector in unrealistic conflict situation. For example two aircraft can enter a sector at the same time at the same level. To limit the recording of unrealistic workloads the maximum number of radar interventions recorded for one flight crossing one sector can be limited (parameter **MAXINT**). It is assumed that in such circumstances, the controller can solve several conflicts situations. A typical value is three. When this value is reached, a single supervision will be recorded instead of an intervention for the new interventions detected. The same logic is applied for the radar supervisions (parameter **MAXSUP**) with a typical value of six.

### Long Flight Time

Specifies a flight time in the sector (**TIM**: expressed in seconds). For all flights with a transit time in the sector higher than this parameter, an additional RT message (Task 68) and an additional Planning Conflict Search (Tasks 50) are recorded. A recommended value for this parameter is 15 minutes (900 seconds).

### Skip Flight Time

Specifies a flight time in the sector (**SKP**: expressed in seconds) below which the sector is skipped. The flight is counted in the sector counts but is considered as skipped (No contact on the frequency). If the sector is the first sector of an ACC, it is considered that the controlling sector is the next sector. If not, the flight remains on the frequency of the transmitting sector. All the RT Tasks and Radar Tasks for this flight will be recorded for the controlling sector.

## DESCRIPTION OF ATC TASKS

### Introduction

CAPAN-Like records various ATC Tasks in the following Categories:

- Flight Data Management
- Coordination
- Planning Conflict search
- RT Communications
- Radar Supervisions and Interventions
- Radar Handovers

Compared with CAPAN or RAMS, the number of tasks has been limited. Nonetheless, the most common controller activities have been considered. In almost all the simulation Studies done with Fast Time Simulation Tools, the first overloaded sector controller was the Executive Controller. Although both EC and PC positions have been considered with Capan-Like, the main effort was devoted to the Executive Controller Tasks. Another reason is that a realistic recording of coordination Tasks which are by essence Planning Controller Tasks, would require the input of many parameters difficult to implement in an analytical Model like SAAM.

### Flight Data Management

The Tasks in this Category are mainly Tasks for the Planning Controller and correspond to Strip Handling and Flight data management. For each flight entering a sector, a Task 44 and a Task 45 are recorded. These Tasks are depending on the Flight Plan Processing System of the ACC simulated. The execution times and the position(s) responsible might be different whether or not there is a Strip-less system available.

**Task 44** corresponds to the receipt of the Strip(s) on the sector.

**Task 45** corresponds to the removal of the flight information after the aircraft has left the sector.

For each Flight entering the first sector of a new ACC, a **Task 1** is recorded. This Task is recorded in order to cover the workload for the controller to manually input a non-foreseen flight plan. As this Task represents a very small number of situations, and as it is impossible to specify to the model the flights concerned, the Task is recorded for each flight, but the execution time is adjusted in proportion of the number of flight concerned. For example, if only 2% of flight plans are manually input, an execution time of 2 " will be specified which represents 2% of 2 minutes (2 minutes is generally the time required to cover this type of activity).

### Coordination Tasks

The Tasks in this Category are mainly Tasks for the Planning Controller. The recording of coordination Tasks has been limited with Capan-Like. A realistic capture of the coordinations Tasks would require a large amount of input parameters impossible to envisage for an analytical model without deteriorating the performance. As in most Sectors the first sector controller position to be overloaded is the Executive Controller, the main effort was put on

the ATC Tasks of this position. The Coordination Tasks are systematically recorded by CAPAN-Like. The best way to measure a realistic workload for the Planning Controller is to associate an execution time to these Tasks in proportion of the number of flights in real life. A

**Task 5** Reception of a time and level estimate from a neighboring Unit. (Inter-Centre)

**Task 15** Transmission of a time and level estimate to a neighboring Unit. (Inter-Centre) .

**Task 25** Reception of a time and level estimate from a sector of the same ACC. (Inter-Sector)

**Task 35** Transmission of a time and level estimate to a sector of the same ACC. (Inter-Sector)

**Tasks 6-16 –26-36** are the same as Tasks 5-15 and 25-35, but they are applied for coordinations between superimposed sectors, thus for flights in climb or descent through the floor or ceiling of the sectors concerned. In addition to the coordination Tasks a conflict search Task is recorded (Task 49) at the agreed transfer level.

### Conflict search Tasks

**Task 50** Standard Planning Conflict Search Task, recorded for each flight before it enters the sector. Also, when the flight time in the sector exceeds the sector parameter (long flight time) an additional Task 50 is recorded for the flight in the middle of the sector.

**Task 51** Additional Planning conflict search Task recorded when the actual number within the sector exceeds the parameter `ADDITIONAL_PLANNING_CFS`. The number of additional Tasks 51 recorded is calculated as follow:  $Nb_{51} = (\text{Actual} - \text{Parameter}) / 2$

**Task 49** This Conflict search is recorded when the next sector is above or below the transmitting sector (superimposed sectors) and corresponds to a conflict search Task at the agreed transfer level between the two sectors.

### RT Communications

**Task 66** First call from an aircraft entering the first sector of an ACC.

**Task 67** First call from an aircraft entering a sector and coming from a sector of the same ACC. This Task is the same as Task 66, but it is the same than Task 66, but a different number has been given because the first contact in the first sector of a new ACC is generally more complex (more detailed instructions).

**Task 64** Instructions to change the SSR code when appropriate. This Task is recorded with Task 66. For those sectors, which may be concerned by this Task, it is recommended to input an execution time reflecting the proportion of flights concerned. A typical execution time for this Task is 10". As example if about 20% of flights are concerned in a sector, an execution time of 2" should be specified for the Task.

**Task 69** Report of aircraft when reaching a flight level. This Task is recorded after a climb or descent phase when the aircraft reaches its cleared flight level. First call from an aircraft entering a sector and coming from a sector of the same ACC.

**Task 71** Instructions to aircraft for a climb or descent phase. This Task is recorded when an aircraft changes its status from cruise to climb or descent to initiate a climbing or descending phase.

**Task 68** Report of aircraft. This Task is recorded when the flight Time through the sector exceeds the Long flight Time parameter of the sector. It is considered that for flights with a long transit time in the sector, it is not enough to record only the First and last messages. In

addition, a Task 51 is recorded which corresponds to a quick verification of the exit conditions by the planning controller.

### Radar Supervisions and Interventions

A detailed description of the logic used to record the Radar Supervision and Intervention Tasks is available at Annex1. The conflict detection of SAAM is performed prior to the execution of CAPAN-Like. This conflict search identifies any infringement to a specified separation threshold given to the Module. The SAAM conflict detection module stores the distances between the two aircraft as well as the times, from the beginning of the conflict to the end of conflict. Capan-Like identifies the sector(s) concerned by the conflict, and records the Radar Supervisions Tasks, the Intervention Tasks and the Radar Handovers Tasks when required.

Capan-Like uses the SUP/INT thresholds specified for the sector(s) concerned, and records the appropriate Tasks. These Tasks are specified by conflict Type. (see detail description at Annex-1)

Nine conflict are identified based on the tracks (same direction/crossing/opposite) of the two flights and on their status (cruise/climb/descent).

Example: crossing conflict between two aircraft in cruise - Type 4.

- Supervision Threshold for the sector concerned = 20 NMS,
- Intervention Threshold for the sector concerned = 10 NMS.

Minimum separation between the two aircraft, and identified by SAAM Minsep = 5 NMS.

In this case, Minsep < INTERVENTION < SUPERVISION.

The Model will record a Supervision **Task 94** (90 + Conflict Type) **and** an Intervention **Task 104** (100 + Conflict Type).

### Radar Handovers

Radar conflicts may concern several sectors. When the two aircraft in conflict are transferred to a sector under Radar Supervision or Radar Intervention, radar coordinations are recorded between the transmitting and the receiving sectors.

**Task 80** Radar Handover of two aircraft subject to a Radar Intervention.

**Task 81** Radar Handover of two aircraft subject to a Radar Supervision.

**Task 90** Radar Handover between the first sector of the flight and a non simulated area. This Task is recorded when the minimum separation is identified before the simulated area. It is equivalent to Task 81.

## POTENTIAL DEVELOPMENTS

To improve the quality of the results, some important future developments are required. The most important one is probably the implementation of Iterations in the module. From a reference traffic sample, it should be possible to simulate several iterations and to average the results.

Generate several so6 files by applying variations on the times and on the speed of the aircraft so that various situations (conflicts, sector penetrations) can be measured. This will smooth the peak effects and the averaged results (capacities and workloads) will be improved.

## CONCLUSION

This Tool was designed to improve the existing assessment of sector workloads. Several validation tests have been made on various ACC's already simulated with CAPAN or RAMS. The results obtained are promising and the figures obtained are closed to the figures obtained with the arithmetic simulation. This tool should be used with care as the modification of basic parameters can have a very significant effect on the results. It will never replace a Fast Time Tool like RAMS or CAPAN which have been designed to give a high level of details, but require a high amount of input parameters.

As for any Simulation Tool, it is recommended to simulated a reference Organisation against which the tested organisations can be compared. The relative comparison of figures is always better than an interpretation of absolute figures.

### 2.5.16 Delay

#### Accessed by

[Menu Bar / Analysis / Delay](#)

#### Purpose / Description

To process the delay of flights for a given scenario, in order to respect declared sector capacity. The purpose is to mimic in a more simple way, the slot allocation algorithm that is used in Eurocontrol CFMU. These delays are a corrective mean to prevent flights to penetrate overloaded sectors..

#### Input

- Traffic file (\*.so6)
- Associated Intersection file (\*.t5)
- Capacity file (\*.cap) *optional*
- Tolerance file (\*.tol) *optional*

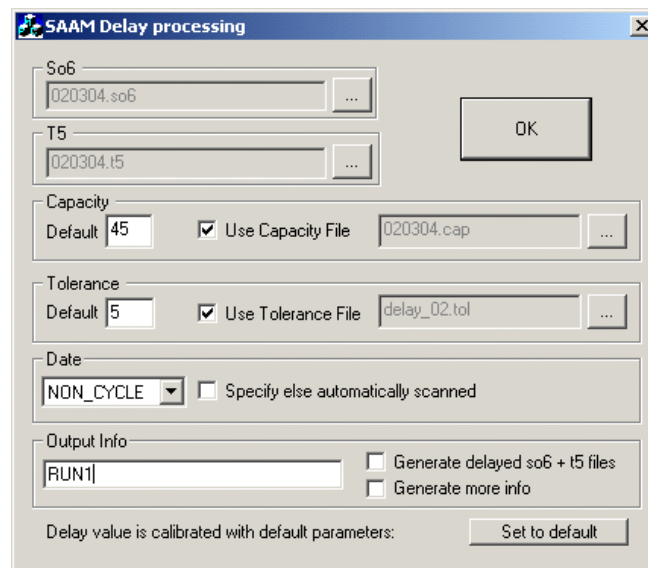
#### Output

- Delayed traffic file (\*.so6) and Airspace/Traffic Intersection file (\*.t5) *optional*
- A text file **\_\_more\_info** *optional*
- A text file **\_\_flights\_bunching**
- A text file **\_\_flights\_capacity**
- A text file **\_\_sector\_delays**
- A text file **\_\_delay\_result**

#### General Procedure

1. Specify input Traffic file
2. Specify associated Intersection file
3. Set a capacity figure or browse a capacity file
4. Set a tolerance figure or browse a tolerance file
5. Select NON\_CYCLE or CYCLE (optional)

6. Specify an output name
7. Click OK



**Delay Dialog Box**

## Parameters

1. **So6 field** – Browse a Traffic file (\*.so6)
2. **T5 field** – Browse an Airspace/Traffic Intersection file (\*.t5)
3. **Capacity frame** – You can either use a default capacity value or you can browse a capacity file. Capacity values, that are read from a file, are systematically increased (internally) by 10% to simulate controller flexibility.
4. **Tolerance frame** – Tolerance of a sector is the number of flights that can be allocated to a slot. You can either use a default value or you can browse a tolerance file.
5. **Date frame** - Here you can select between cycle or non\_cycle.
  - **CYCLE** mode, if no available slot is found for a flight at the end of the day, the search must resume at the beginning of the day.
  - **NON-CYCLE** mode, a flight must be allocated to a slot of the day (it may not be allocated to a slot of the next day) or not be allocated at all.

When **NON-CYCLE** is selected you can tick the check-box to set the date or it will be automatically scanned.

6. **Output Info frame** – An output name has to be written in the field. This name will be given to all the output files.

There is are two check-boxes.

- **Generate delayed so6+t5 files** – When ticked will generate a Traffic file and an Airspace/Traffic Intersection file, updated with the delay.
- **Generate more info** – Generates a file containing information about delay imposed on each flight  
**Flight\_id      crossed\_sectors      delay\_value\_in\_seconds      delay\_type(C for Capacity, B for Bunching, x means no delay)**

Seven files are expected to be created after the run of the Delay processing module:

- The file **\_\_flights\_bunching**, listing each flight, its bunching delay and the name of the sector responsible for the delay format is:

Flight_id	origin	destination	most_bunching_penalising_sectors
total_delay_value(=capacity+bunching)			

- The file **\_\_flights\_capacity**, listing each flight, its capacity delay and the name of the sector responsible for the delay

Flight_id	origin	destination	most_capacity_penalising_sectors
total_delay_value(=capacity+bunching)			

- The file **\_\_sector\_delays**, listing the bunching, capacity and total delays generated by each sector. Delays are expressed in seconds
- The file **\_\_delay\_result**, containing the parameters passed to the module, the processing time, the total bunching delay and the total capacity delay (summed on all flights) in minutes.

Note the difference between delays and ADJUSTED delays. The real delay that are processed are generally too high. By comparing delay figures provided by CFMU and delay figures calculated by SAAM for several days, we found that a ratio could be applied. This ratio was embedded in the processing module and gives the ADJUSTED delay.

- A file containing warnings and errors. This file is found in the “cache” directory

**N.B.:** All output files are all providing delays in seconds unless it is written "minutes".

## 2.5.17 Airspace PRU Complexity

### Accessed by

[Menu Bar / Analysis / Airspace PRU Complexity](#)

### Purpose / Description

Calculates the complexity of an airspace based on several indicators, traffic density, traffic in evolution, flow structure and traffic mix. The calculations and indicators are based on PRUs report about Complexity Metrics for ANSP and was developed by French DSNA in 2006.

### Input

- Traffic file (\*.so6)
- Airspace files (\*.are and \*.sls)

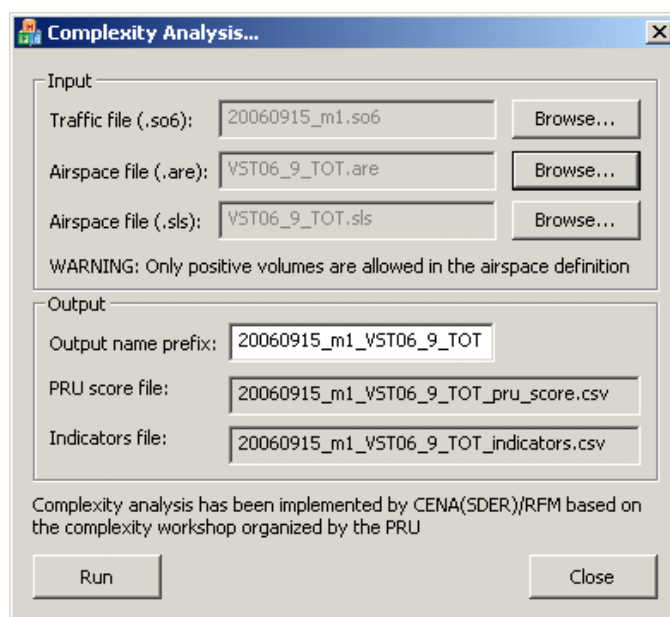
### Output

- PRU score file (\*.csv)
- Indicators file (\*.csv)

### General Procedure

1. Browse input Traffic file
2. Browse Airspace files
3. Specify an output name or accept the proposed one
4. Click RUN





**Airspace PRU Complexity Dialog Box**

## Parameters

### 1. Input frame -

- a) **Traffic file (.so6) field** – Browse a Traffic file
- b) **Airspace file (.are) field** – Browse an Airspace file
- c) **Airspace file (.sls) field** – If a \*.sls file is found with the same name as the browsed \*.are file, it is automatically selected. The airspace file should be made with areas enough large (like size of an ACC) to be covered by enough grid cells (generated internally to calculate indicators).

### 2. Output frame –

- a) **Output name prefix field** – An output name prefix is suggested, traffic\_file\_airspace\_file. You can either accept this or type another one.
- b) **Indicators file field** - An excel file providing 11 following columns, from which you will find 10 elementary indicators:
  - name of your area
  - Nb cells: number of crossed cells in your area
  - Sum FL\_i: sum of crossed cells DFL (expressed in FL)
  - Sum T\_i: sum of flight time (in minute)
  - Sum T\_i^2: sum of square of flight time (square minute)
  - Sum NM\_i: sum of distances (in NM)
  - Sum NM\_i^2: sum of square of distance (in square NM)
  - Sum HC\_i: number of horizontal crossing
  - Sum VC\_i: number of vertical crossing
  - Sum EtV\_i\*N\_i: standard deviation of speed averaged by N\_i
  - Sum N\_i: sum of number of flights crossing the cells
- c) **PRU score file field** – An excel file providing composite indicators (calculated from elementary indicators) and the final complexity score.
  - Adjusted Density (AdjD) =  $\text{Sum } T_i^2 / \text{Sum } T_i$

- Horizontal Different Interactive Flows (HDif), is a measure of the complexity arising from the interactions between flights with different headings =  $\text{Sum HC}_i / \text{Sum NM}_i$
- Vertical Different Interactive Flows (VDif), is a measure of the complexity arising from the interactions between flights in different phases =  $\text{Sum VC} / \text{Sum NM}_i$
- Traffic Mix (TMix), is a measure of the complexity arising from the interactions between flights with different speeds =  $\text{Sum EtV}_i * \text{N}_i / \text{Sum N}_i$
- Complexity Score (Score), is the relative (compared to your areas set) complexity score for each area. It is the sum of the 4 standardized composite indicators value. A standardized value is the value (for an area for a composite indicator) minus the average divided by the standard deviation.  $\text{Standardized value} = (\text{value} - \text{average}) / \text{stddev}$ . The average and the standard deviation (non biased !!) being calculated with all values (of all areas) from a given composite indicator. The score has always an average of 0 (zero), so a positive value is relatively more complex than the average, and a negative value is relatively less complex than the average. With the term "relative" means for your set of areas ! Note that other implementations of the PRU complexity indicators could use a slightly different way to aggregate the composite indicators (for instance it could be the average of the 4 normalized values of the composite indicators, etc ...).

### Important notes:

- The grid coverage generated internally is running: in latitude from W24 to E35, in longitude from N19 to 72N, in FL from FL85 to FL445.
- Excel files which are generated might use different decimal separator than your system, and the field (column) separator might be different as well. So if you need to manipulate numbers provided in these Excel sheets, check this first !

## 2.5.18 Airspace Flight Time (minutes)

### Accessed by

[Menu Bar / Analysis / Airspace Flight Hours](#)

### Purpose / Description

To process total flight time in minutes and number of flights for each airspace present in the input airspace files, \*.are and \*.sls. This is done for one or several days, depending on the number of input traffic files, \*.so6. The output file contains results for all the airspace and all days of traffic. Name of the output file is, name of the first traffic, name of the last traffic and "\_flighthours.txt and it is comma separated.

### Input

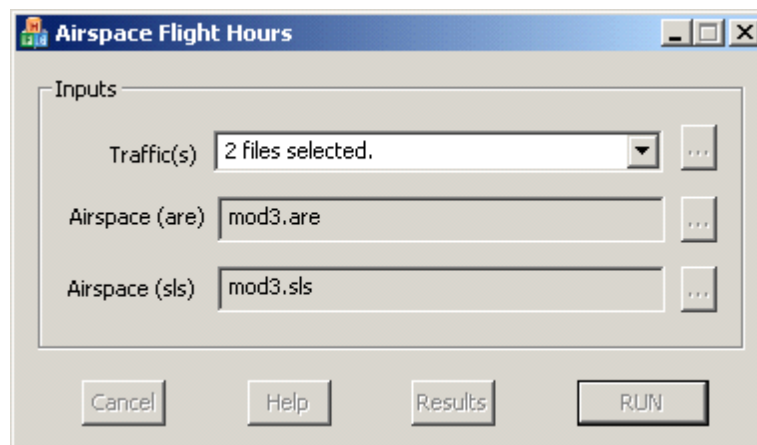
- One or several Traffic file/s (\*.so6)
- Airspace files (\*.are and \*.sls)

## Output

- Text file “first traffic\_last traffic\_flighthours.txt

## General Procedure

1. Browse one or several traffic file/s (\*.so6)
2. Browse Airspace files, both \*.are and \*.sls
3. Click RUN



**Airspace Flight Hours Dialog Box**

## Parameters

1. **Inputs frame** -
  - a) **Traffic(s) field** – Browse a one or several Traffic file/s. All your traffic file must be located in the same directory
  - b) **Airspace (are) field** – Browse an Airspace file, \*.are
  - c) **Airspace (sls) field** – Browse an Airspace file, \*.sls. If there is a \*.sls file with the same name as the browsed \*.are file, it will automatically be selected.
2. **Cancel button** – Closes the dialog box
3. **Help button** – Opens a new window with help
4. **Results button** – Opens the working directory
5. **RUN button** – Starts the process

### 2.5.19 Scenario Economy

#### Accessed by

[Menu Bar / Analysis / Scenario Economy](#)

#### Purpose / Description

Compare length, time, fuel, CO<sub>2</sub> and NOx emission of two traffic files. All flights present in both files where some airspace changes were applied are processed.

The two input traffic files, the reference and the scenario, are generally taken from the same day of traffic. The flights are identified with their flight ID. Only the flights present in both input files are analysed. The TMA exclusion is optional, if not selected, analysis will be done from origin to destination.

As usual, it is strongly recommended to build your scenario (=after situation) on a clean and stable reference (=before situation), so that the comparison between the reference and the scenario will fairly reflect your changes and only your changes.

## Input

- Reference Traffic file (\*.so6)
- Scenario Traffic file (\*.so6)

## • Output

Three output files are generated:

- [The result file](#)
- [The suspicious file](#)
- [The missing flights file](#)

## The result file

The excel file "ScenarioEconomic.xls" (default name) contains the results for the valid flights.

The comparison is performed on five criteria: Length, Time, Fuel, CO<sub>2</sub> and NOx. A flight appear in the output only if a change has been detected on at least one criteria. A change is detected when the absolute value of the delta between the reference and the scenario for a criteria is greater or equal than the threshold defined for this criteria. It means that a flight that doesn't change between the reference and the scenario doesn't appear in the output.

**Note:** The thresholds can be changed from the dialog box.

The file shows the following information for each flight from left to right:

### 1. Fight information:

- Flight ID (unique ID coming from PRISME database)
- Origin (ICAO code)
- Destination (ICAO code)
- Aircraft Type (ICAO code)
- Callsign (as found in flight plan)

### 2. Result (Delta between 3. Reference and 4. Scenario) for all individual flights, plus the grand total (located at the bottom). A positive value means an increase passing from Reference to Scenario.

The result will show for each pair flight, the difference in:

- Length (NM),
- Time (minute),

- Fuel (kg),
- CO<sub>2</sub> (kg),
- NO<sub>x</sub> (kg).

3. Reference: the same 5 fields : length(NM), time(mn), fuel consumption(kg), CO<sub>2</sub> emission(kg), NO<sub>x</sub> emission(kg)

4. Scenario: the same 5 fields : length(NM), time(mn), fuel consumption(kg), CO<sub>2</sub> emission(kg), NO<sub>x</sub> emission(kg)

At the end of the file the *global results* are presented. For each criteria you get the following values:

- *INCREASE*: The total increase value for the criteria and the number of flight concerned by this increase.
- *EQUAL*: The total number of flight which has no significant changes on this criteria (but which has changes on at least one other criteria)
- *DECREASE*: The total decrease value for the criteria and the number of flight concerned by this decrease.

**Note:** This process is using a model derived from AEM model and data. This is a simplified model aimed at providing reliable and consistent results regarding airspace changes and reference/scenario comparisons. The results are not supposed to be taken for granted as far as absolute figures are concerned.

### The suspicious flights

The excel file "*\*\_SuspiciousFlights.xls*" is formatted in the same way as the main output file. It contains flights which have been laid aside because their results needs to be checked. However, they are analysed in the same way as "normal" flights.

The cause of suspicion for a flight is due to a difference of route length in the zone (or from origin to destination if the zone was not selected) which is N times greater or N times smaller. N is the value that can be changed in the dialog box, by default it is set to 2.00. So by default flights having a difference of route length extension 2 times bigger or 2 times smaller will be considered suspicious, and should be analysed.

### The missing flights

The file "*\*\_missingFlights.txt*" contains the list of flights which have been discarded because of missing aircraft information. The number of discarded flights is also written in the main output file.

## General Procedure

1. Browse reference traffic file (\*.so6)

2. Browse scenario traffic file (\*.so6)
3. Possibly tune the TMA exclusion size
4. Possibly tune the comparison threshold values
5. Possibly tune the value for suspicious flights
6. Click Run
7. Once finished you can access to the Excel sheet contains the result by clicking the "results" button.

**Scenario Economy Dialog Box**


## Parameters

1. **Inputs frame** -
  - a) **Reference (so6) field** – Browse the reference traffic file (\*.so6)
  - b) **Scenario (so6) field** – Browse the scenario traffic file (\*.so6)
  - c) **TMA checkbox** – Enables TMA route part automatic exclusion before reference/ scenario flights comparison. Browse an Airspace file, \*.are
  - d) **TMA size value** – *Optional* – Specify the radix in NM used to draw a virtual and circular TMA around departure and arrival airports for each flight. Then, reference/ scenario comparison is performed on the portions of the flights that lie outside these TMAs, as it is done for [Route length comparison](#).
2. **Comparison thresholds frame** - Change the threshold value for each criteria. A flight appear in the output only if a change has been detected on at least one criteria. A change is detected when the absolute value of the delta between the reference and the scenario for

a criteria is greater than the threshold defined for this criteria. It means that a flight that doesn't change between the reference and the scenario doesn't appear in the output.

- a) **Length value** - The length threshold in nautical miles
- b) **Time value** - The time threshold in seconds
- c) **Fuel value** - The fuel threshold in kilograms
- d) **CO<sub>2</sub> value** - The CO<sub>2</sub> threshold in grams
- e) **NOx value** - The NOx threshold in grams
- f) **Keep default proportion checkbox** - If checked, changing one threshold will automatically change the other ones in the default proportion.

### 3. Output frame -

- a) **Suspicious flights (value)** – The value can't go below 1.00. For instance 1.5 means limit is 50% increase of decrease in delta length extension. If the suspicious flights checkbox is disabled, it means that no flights will be dumped. The results for these flights are stored in "ScenarioEconomic\_SuspiciousFlights.xls".
- b) **Excel output (xls) field** – Choose the name and location of the output file. The default value is "ScenarioEconomic.xls". Click on the  button in order to open the result in excel.

4. **Run button** – Starts the process

6. **Close button** – Closes the dialog box

## 2.5.20 Airspace Value Merge (Complexity)

### Accessed by

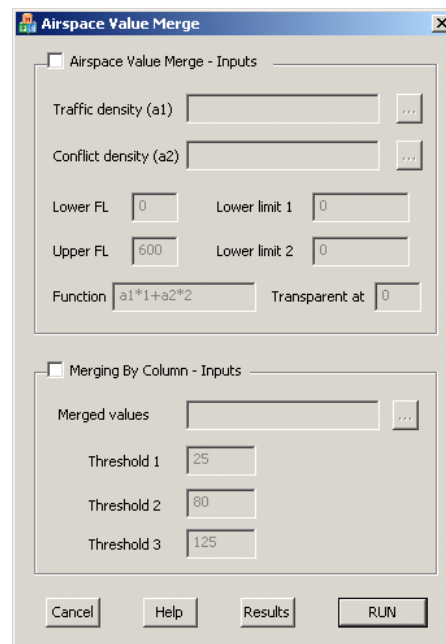
**Menu Bar / Analysis / Airspace Value Merge (complexity)**

### Purpose / Description

This function is used to put together (merge) a traffic density with a conflict density (the two \*.are files must have the same size) and to process and display a map of "high complexity area", useful to design future FAB sector family. This function provides users 2 independent steps:

A way to merge the values for each cell of two 3D densities files (identical) into a single density file. The output file is simply called "merged.are". A formula can be input to tailor the way the merge will be done, by default it is: density value \* 1 + conflict value \* 2 (expressed with  $a1*1 + a2*2$ ). Possible function that can be used in the formula are: sqrt(value) for square root, exp(value) for exponential and log(value) for natural logarithm. As an example it could be:  $\log(a1)*1.2 + \exp(a2)*3.5$ .

Additionally, in a second step, a facility to merge all cells of a column into a single cell (so it goes from 3D densities to 2D densities). This single cell per column will be coloured linearly based on its value according to user's thresholds. By default the output of the first step ("merged.are") is used as an input for the second step. The output file is an .ARE file called with the input file followed by "Column.are" (so by default it is "mergedColumn.are"). An output file called "mergedColumn.txt" is also created and contains information for each cell.



**Airspace Value Merge Dialog Box**

## 2.6 Add Menu

### Accessed by

**Menu Bar / Add**

### Purpose / Description

To add and visualise into SAAM :

- [Airspace\(s\)](#)
- [Network\(s\)](#)
- [Terrain data](#)
- [Runway\(s\)](#)
- [Traffic animation\(s\)](#)
- [Circle\(s\)](#)

This operation can be repeated several times, even for the same type of object, hence the term 'Add'. As an example you can add a current network, a future network and the European sectors to the same SAAM viewport.

#### Notes:

Additional objects such as prisms, lines, texts, images can be added to SAAM viewport. There is no GUI (dialog boxes) to achieve this: you must edit the TDV file and add specific commands: see [Adding graphical objects](#).

In addition to Traffic animation, you can create animations of other objects (shapes, text ...). There is no GUI (dialog boxes) to achieve this: you must edit the TDV file and add specific



commands: see [Creating a SAAM animation](#)

To remove objects from the SAAM viewport, there is no GUI: you must [edit your TDV file](#). Simply remove the command that relates to the object you want to remove or put a # in front of the command.

## 2.6.1 Add Airspace

### Accessed by

[Menu Bar / Add / Airspace](#)

### Purpose / Description

'Add Airspace' let you import into 'SAAM' an Airspace file (\*.are) and visualise them as 3D volumes. This operation can be repeated several times hence the term 'Add'. It is possible to determine a min. and max. flight level for each volume as well as a colour and a transparency factor. **DO NOT** forget to save your modifications (using the 'Save' item on the main menu bar, otherwise they will be lost.

### Input

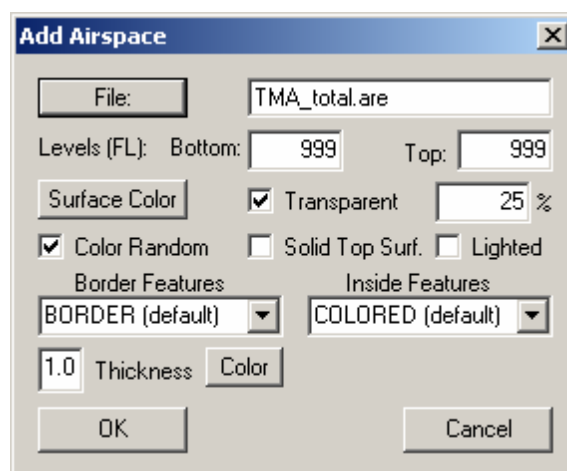
- Airspace file(s) \*.are

### Output

- The \*.are file will be added to the TDV file when saved. Modifications to the airspace volumes will also be saved.

### General Procedure

1. Specify input Airspace file (\*.are)
2. Modify the airspace volume (*optional*)
3. Click OK to display.



**Add Airspace Dialog Box**

## Parameters

1. **File:** *field* – Browse one or several airspace file/s
2. **Levels (FL):** *field* - To specify the Bottom and Top level of the airspace volume
3. **Surface Color** *button* – Changes the colour of airspace volume
4. **Transparent** *check box* – Displays the airspace volume transparent.
5. **%** *field* - Decides level of transparency
6. **Color Random** *check box* – If the airspace file consists of more than one airspace volume a different colour be given to each volume.
7. **Solid Top Surf.** *check box* – Displays the top of the airspace volume solid.
8. **Lighted** *check box* – The top surface can be lighted or not (the colour will change slightly depending on the tilt of the picture)
9. **Border Features** *drop-down list* –
  - a. BLACK (default)
  - b. COLORED
  - c. NOT DISPLAYED
10. **Inside Features** *drop-down list* –
  - a. COLORED (default)
  - b. NOT DISPLAYED
11. **Thickness** *field* – To change the thickness of border. There is Color button to change the colour of the border.

### 2.6.2 Add Network

#### Accessed by

Menu Bar / Add / Network

#### Purpose / Description

The network format is a \*.ase file. It can be either imported, created from scratch or copied from a previous version

#### Input

Network file(s) \*.ase to import and visualise network

#### Output

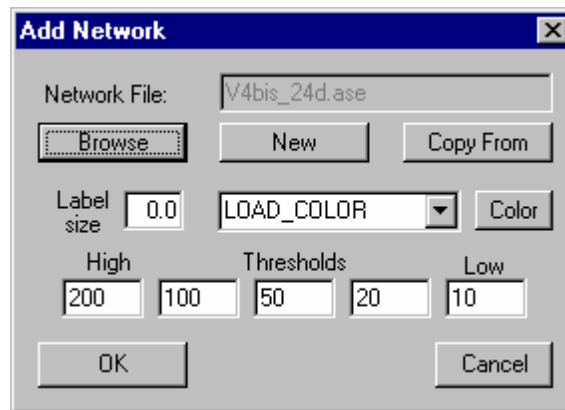
Displays the network and adds it to the TDV file.

#### General Procedure

Choose one of the following options:

- Add an existing network
  1. Specify network file (\*.ase)
  2. Modify the display (*optional*)
  3. Click OK.
- Create a new network

1. Click on new
  2. Browse the location and type the name of the new network.
  3. Modify the display (*optional*)
  4. Click OK.
- Copy from a previous version
    1. Click on '**Copy From**' to duplicate an already existing network file.
    2. Modify the display (*optional*)
    3. Click OK.



**Add Network Dialog Box**

## Parameters

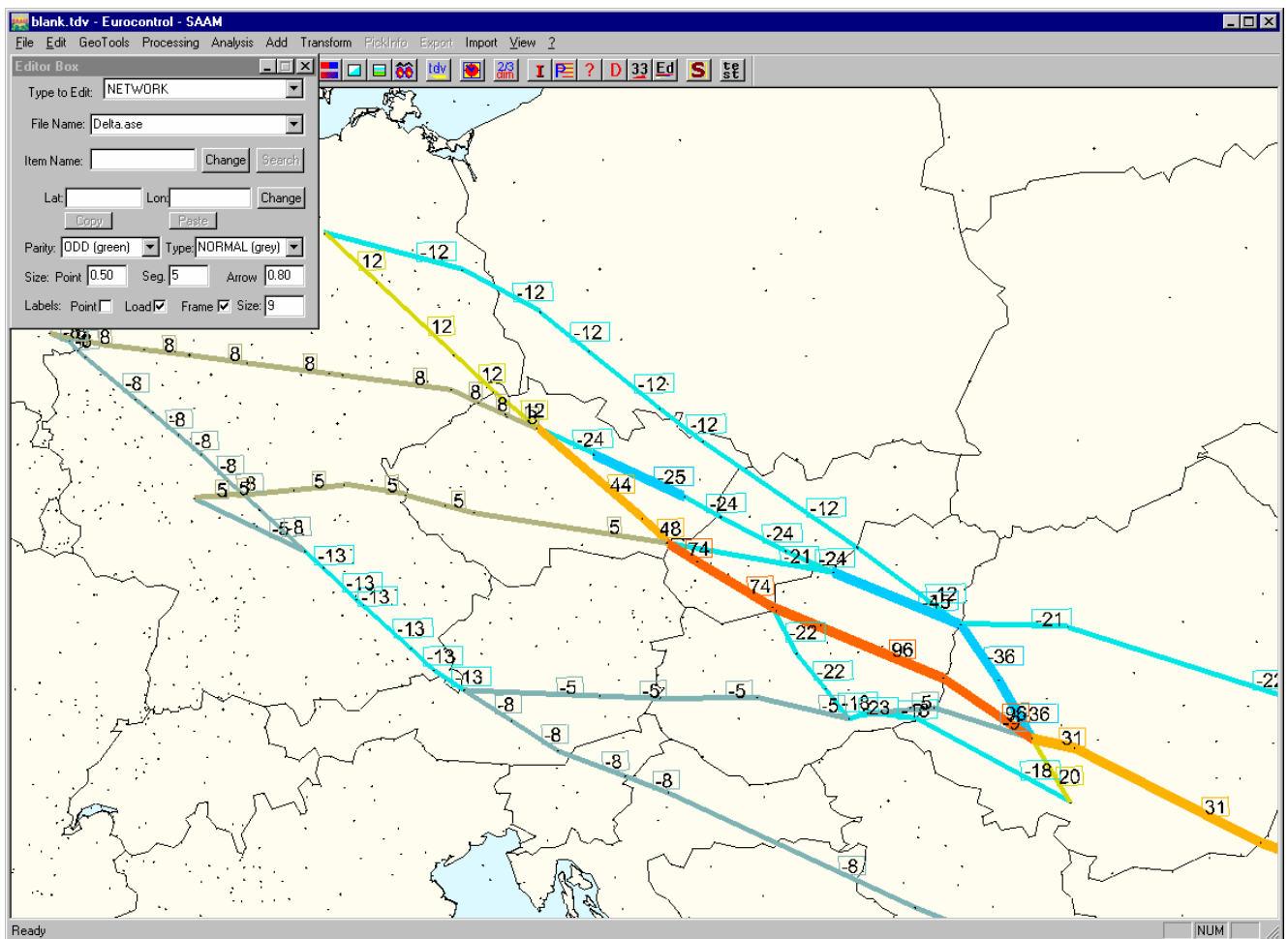
1. **New button** – Creates a new \*.ase file in your working directory called 'new\_network.ase'.
2. **Copy From button** – Duplicates an already existing network file
3. **Label size field** – Changes the size of the labels.
4. **drop-down list** – See [Editing Network](#)
  - DEFAULT\_COLOR
  - PARITY\_COLOR
  - LOAD\_COLOR
  - TYPE\_COLOR
  - MIXED\_COLOR
  - CDR\_COLOR
5. **Color field** – Sets the default colour.
6. **Thresholds fields** – To change the threshold from **High** to **Low** to get a satisfactory display : differences in load are usually between 100 and 0.

### N.B.:

Use the facilities provided in the [Network Editor](#) if you want to change the size of the labels.

Adjustments in the width of the segments and display of the figures are usually necessary . This is done via the [Network Editor](#)

A good display would be like this :



Example of a good display

## 2.6.3 Add Terrain

### Accessed by

[Menu Bar / Add / Terrain](#)

### Purpose / Description

Lets you import and view terrain data. Warning: you must use your own terrain data since SAAM does not provide such data.

### Input

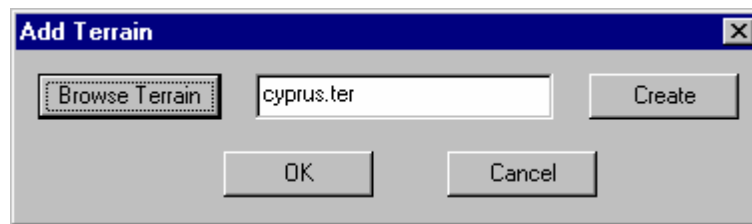
- Terrain data ([\\*.ter](#)) to import and visualise

### Output

- Displays the terrain and adds it to the TDV file.

## General Procedure

1. Select a Terrain file (\*.ter)
2. Click **OK** to display



**Add Terrain Dialog Box**

## Parameters

1. **Browse Terrain** *button* – Browse a terrain data file
2. **Create** *button* -

### 2.6.4 Add Runway

#### Accessed by

[Menu Bar / Add / Runway](#)

#### Purpose / Description

Runways can be added:

- via a dialog box, using runway data provided by SAAM - Warning: this data is old and might not be up-to-date
- manually by editing the TDV and entering your own data

#### Adding a runway via a dialog box

##### Input

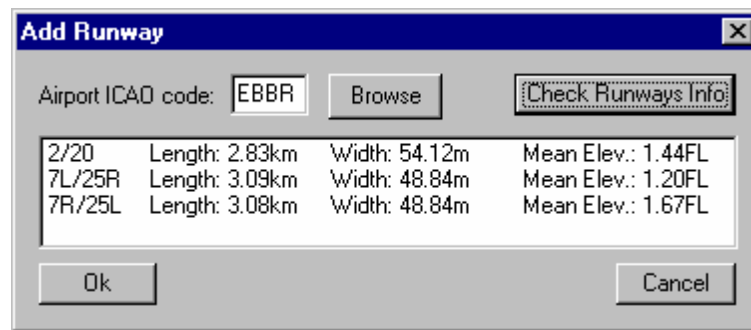
- ICAO code of the airport

##### Output

- Displays the runway and adds it to the TDV file

#### General Procedure

1. Type in the ICAO code of the airport or select it from the list (click on '**Browse**').

**Add Runway Dialog Box**

### Parameters

1. **Airport ICAO code field** – Type in the ICAO code of the Airport
2. **Browse button** – Opens a list of ICAO codes
3. **Check Runways Info button** – Gives you information about the runways at the selected airport.

### Adding a runway by editing the TDV

To add a runway based on your own data or to correct the SAAM runway data, you must [edit your TDV](#) manually with a text editor. The **RUNWAY** TDV command has the following syntax:

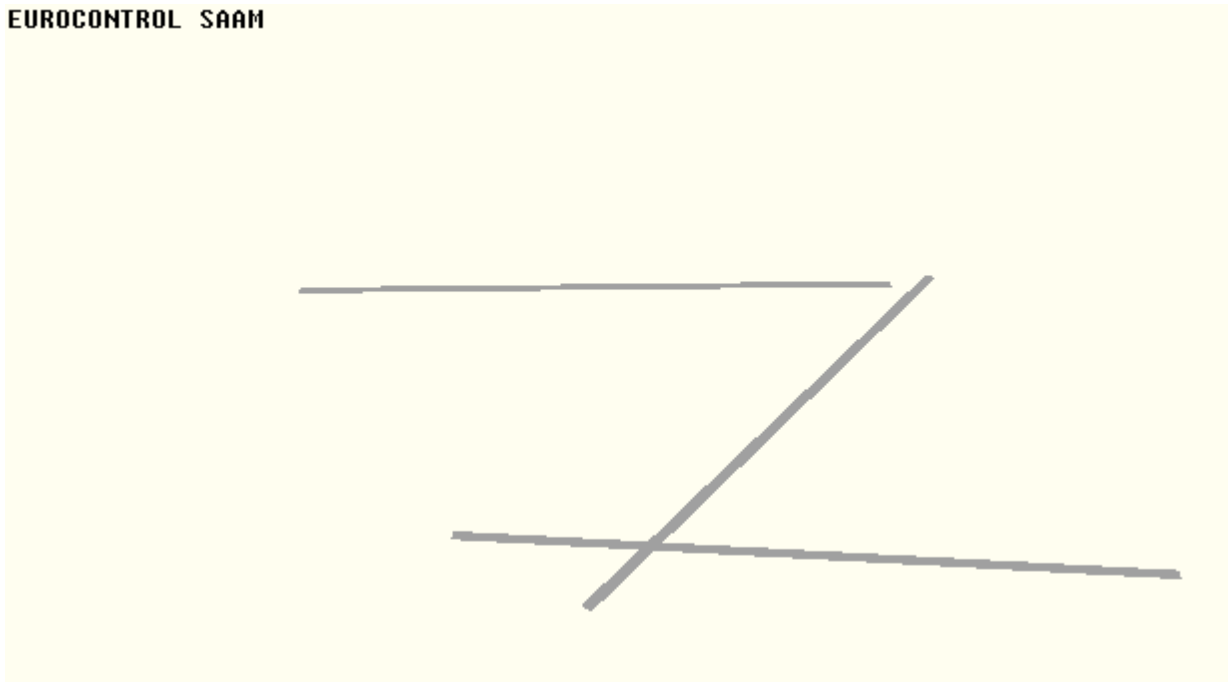
Syntax:

```
RUNWAY airport_name runway_name lat1 lon1 elev1 stop1 lat2 lon2 elev2 stop2 width COLO
```

Examples:

```
RUNWAY EBBR 2/20 3053.300000 269.517000 1.800000 0.000000 3054.780000 270.117000 1.070
RUNWAY EBBR 7L/25R 3054.000000 267.567000 1.290000 0.000000 3054.700000 269.967000 1.1
RUNWAY EBBR xxx 3053.370000 268.917000 1.750000 0.000000 3053.930000 271.400000 1.5900
```

EUROCONTROL SAAM



EBBR 3 runways.

#### ▣ Airport parameters

**airport\_name:**

**runway\_name:**

**lat1 lon1 elev1:** coordinate of one end of the runway (elev in FL)

**stop1:**

**lat2 lon2 elev2:** coordinate of the other end of the runway (elev in FL)

**stop2:**

**width:** width of the runway (in nm)

**COLOR:**

## 2.6.5 Add Traffic Animation

### Accessed by

[Menu Bar / Add / Animation](#)

### Purpose / Description

An traffic animation is basically a time-related display of a traffic sample: the position of each aircraft in the sample is shown at regular time-intervals.

**Note:** You can also add animations of non traffic object in SAAM: animation of a sector, a network, images, titles.... For further information, please refer to [Creating a SAAM animation](#)

Several animations (of same or different traffic) are possible with different aircraft shape or color ...

## Input

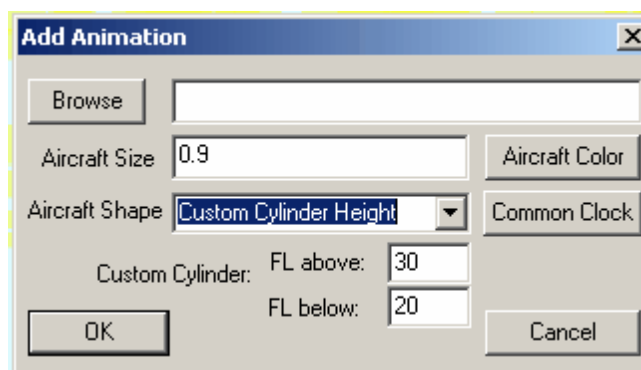
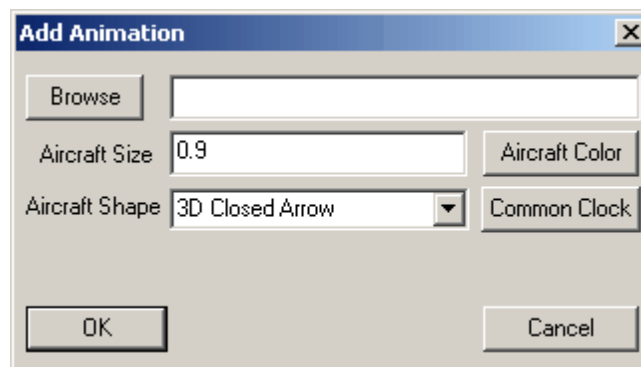
- A traffic file (\*.so6)

## Output

- Animation added to the TDV file.

## General Procedure

1. Click on '**Browse**' to open a \*.so6 file.
2. Modify the aircraft shape, size and colour (*optional*)
3. Set the '**Common Clock**'.
4. Click on '**OK**' to run. The computing time may take several minutes.

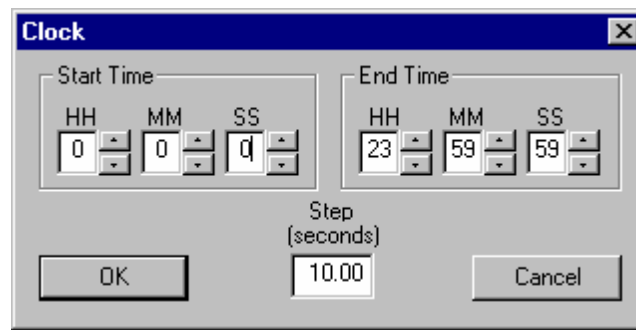


**Add Animation Dialog Box**

## Parameters

1. **Browse button** - Browse a traffic file
2. **Aircraft Size field** – To specify the size of the aircraft in pixel or nautical miles. Nautical miles are used for big shapes, cylinders and noise shadow. Type XXnm or XXNM with XX numeric value.
3. **Aircraft Color button** – To change the colour of the aircraft.
4. **Aircraft Shape drop-down list** – Gives you a selection of different aircraft shapes.
5. **Common Clock button** – Serves to specify the time intervals.







**Common Clock Dialog Box**

## Parameters

1. **Start Time fields** – Used to define the start time
2. **End Time fields** - Used to define the end time
3. **Step (seconds) field** – To specify the time intervals used to display the positions of the aircraft. The usual values are between 5 and 60 seconds.

## N.B.

Once the animation has started, some parameters (start and end time, speed) can still be changed.

- The speed of the animation can also be changed by using the ‘2’ and ‘8’ numerals on the numerical pad. ‘2’ halves the animation speed whereas ‘8’ increases it by a factor of two.
- Click on the ‘**Clock**’ button ,  , in the Toolbar to change the animation clock.
- By clicking on the ‘**Aircraft Label**’ button ,  , in the Toolbar you can view the flight levels and the flight ID during animation. All other options are not yet implemented.

By clicking on the Pause/Break button the animation can be paused and continued again (it freezes/unfreezes the clock).

After you have saved your animation, you can open the TDV file. There you will see two lines that have been added for the animation. One line starts with **CLOCKx** and the other one starts with **SANIM**. At the end of the **CLOCKx** line there are three letters **NTS**, that stands for Number of Aircraft, Time and Speed. This is the information you’ll get at the bottom left on view port when the animation is running. You can decide not to view this information by replacing each letter with an ‘X’.

### 2.6.6 Add Circle

#### Accessed by

[Menu Bar / Add / Circle](#)

#### Purpose / Description

Will draw a perfect Circle, using the ‘**Add Circle**’ option. The ‘SLS’ option allows the programme to define a corresponding \*.sls file which is necessary if the event of [Airspace](#)

[Load](#) or [Airspace/Traffic Intersection](#) calculation.

## Input

- Type in relevant parameters.

## Output

- An airspace file (\*.are)

## General Procedure

1. Type in relevant parameters
2. Click on 'OK' to generate the output file.

**Add Circle Dialog Box**

## Parameters

1. **Center fields** – To specify the position of the circles centre in Lat/Long.
2. **Locate button** – Not implemented
3. **Radius (nm) field** – To specify the radius of the circle in Nautical miles.
4. **Number of Points field** – To define number of points in the circle. The more point the more perfect round circle.
5. **Azimuth (degree) field** – Not implemented
6. **Flight Levels fields** – To determine the **Bottom** and **Top** level of the circle.
7. **Name field** – Name of the output file. Default is NEW\_CIRCLE\_40.are
8. **Colour button** – Opens a colour palette were you can change the colour of the circle.
9. **Trans. checkbox** – An option to display the circle in transparent.
10. **Generate SLS checkbox** – Generates a corresponding \*.sls file which is necessary in the event of Airspace Load/Intersection calculation.
11. **name: field** - An output name is suggested. It can be changed.

## 2.6.7 Add LayerSet

### Accessed by

**Menu Bar / Add / LayerSet**

### Purpose / Description

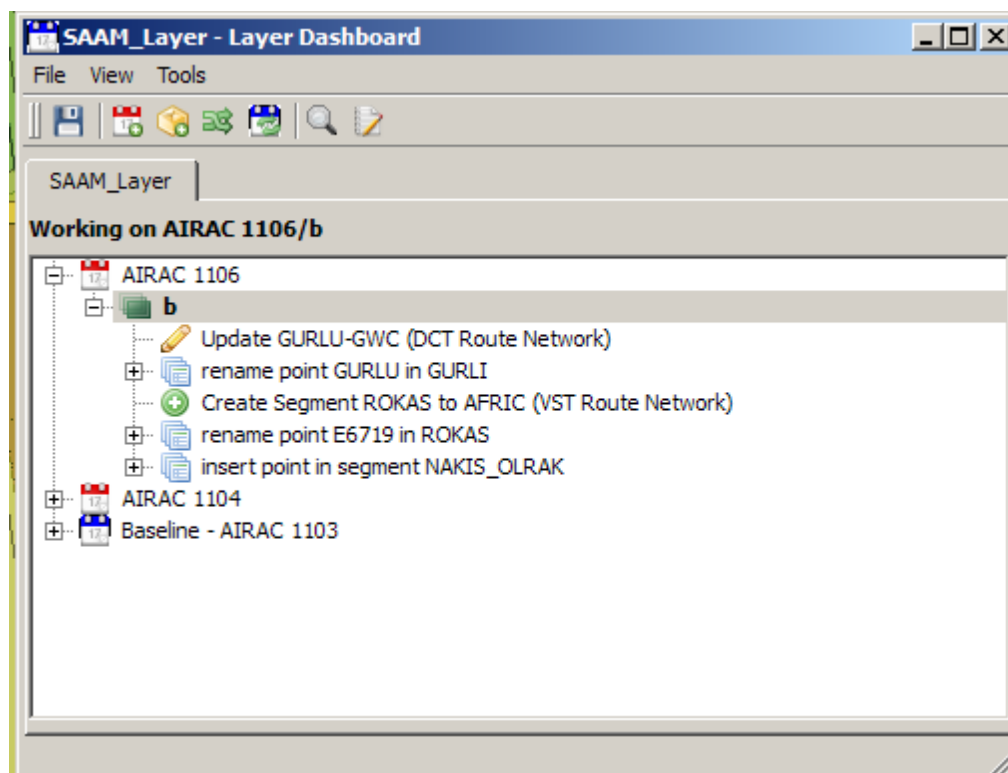
This function is linked to Layer which is provided for internal test.

This "Add LayerSet" function activate, if possible, the Layer mechanism implemented in SAAM, or re-display the LayerDashboard if closed by user.

Layers, when activated, allows to stored any changes of airspace structure, using classical SAAM editors, per project, each project being stored within an AIRAC date. When editing, horizontal consistency check (consistency between different files for a given view) and vertical consistency check (consistency between the different projects) are launched automatically, and users might be prompted to manually correct inconsistencies.

Layers provide an automatically compilation and recomputation of a complete Dataset in your working directory (the files of this dataset are flagged with AIRAC number "0000") based on a simple click on the tree presenting projects in the LayerDashboard. This computation takes the baseline files (original dataset) and add to it all modifications stored in each project in a time line ordered way from baseline to the project select as the basis of your view.

Layer is only currently available for people within EUROCONTROL as it uses, for the time being, a shared folder. Later on, it is planned to replace this shared folder with a SVN server providing access to Layer to external users.



**LayerDashboard with 2 AIRAC cycles - Project "b" and all projects below being activated.**

Different editors are Layer compliant:

- Network editor (classical one)
- new RAD editor

## 2.7 Transform Menu

### Accessed by

**Menu Bar / Transform**

### Purpose / Description

This is a tool for transforming one file format into another file format.

### 2.7.1 S06 to Text

#### Accessed by

**Menu Bar / Transform / So6 to.../ So6 -> Text**

#### Purpose / Description

This menu allows the user to extract data from a so6 file in a field format, which can then be imported into Excel or a text editor.

#### Input

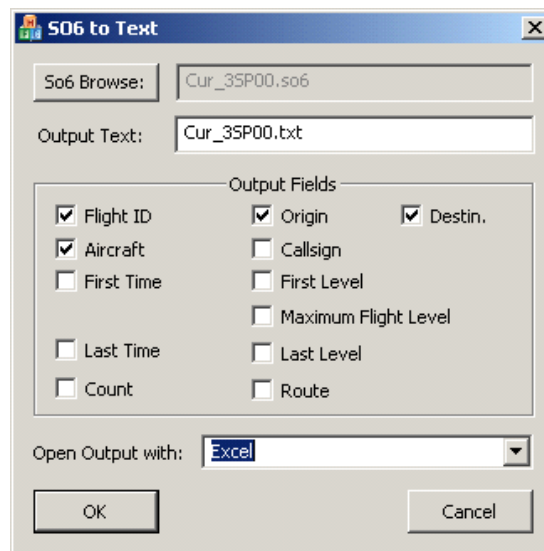
- Traffic file (\*.so6)

#### Output

- Text file (\*.txt)

#### General Procedure

1. Specify the input Traffic file
2. Allocate a name to the output text file
3. Select fields to be extracted
4. Select Excel or Text editor to display text file
5. Click OK to Run



**So6 to Text Dialog Box**

## Parameters

1. **So6 Browse field** - Browse traffic file
2. **Output Text field** - An output name is suggested, same as input. Can be changed.
3. **Output Fields checkbox options** – To select which fields to extract from the input file.
4. **Open Output with drop-down options** – Determines which program to open the output file with. The options are **Text Editor**, **Excel** or **NO VIEW**.

## Note

The field separator is a TAB.

The **order of the fields** in the result file is fixed as follows: 0: Flight id, 1: Origin, 2: Destination, 3: Aircraft Type, 4: Callsign, 5: First Time, 6: First Level, 7: Max Flight level, 8: Last Time, 9: Last Level, 10: Count, 11: Route.

### 2.7.2 SO6 to Traffic Demand ( .exp )

#### Accessed by

**Menu Bar / Transform / So6 to ... / So6 -> Traffic Demand ( .exp )**

#### Purpose / Description

Converts an existing traffic file (\*.so6) to a traffic demand file (\*.exp). Which is used as an input to the [Assignment](#) process.

**Note:** This function is here to insure backward compatibility. It is recommended to download from DDR traffic demand in the form of \*.exp2 (more information are stored in an \*.exp2 facilitating conversion to ALLFT files. Especially, RFL lying in exp2 files on DDR is more exact than the one interpreted from So6 data.). So there is no need to convert an \*.so6 traffic

into \*.exp traffic demand.

## Input

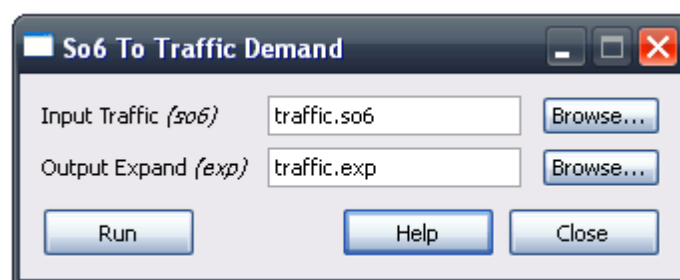
- Traffic file (\*.so6)

## Output

- Traffic Demand (\*.exp)

## General Procedure

1. Specify the input Traffic file
2. Allocate a name to the output file
3. Click Run



**So6 To Traffic Demand Dialog Box**

### 2.7.3 S06 to Network

#### Accessed by

[Menu Bar / Transform / So6 to ... / So6 -> Network](#)

#### Purpose / Description

Converts an existing traffic file (\*.so6) to a network file (\*.ase).

## Input

- Traffic file (\*.so6)

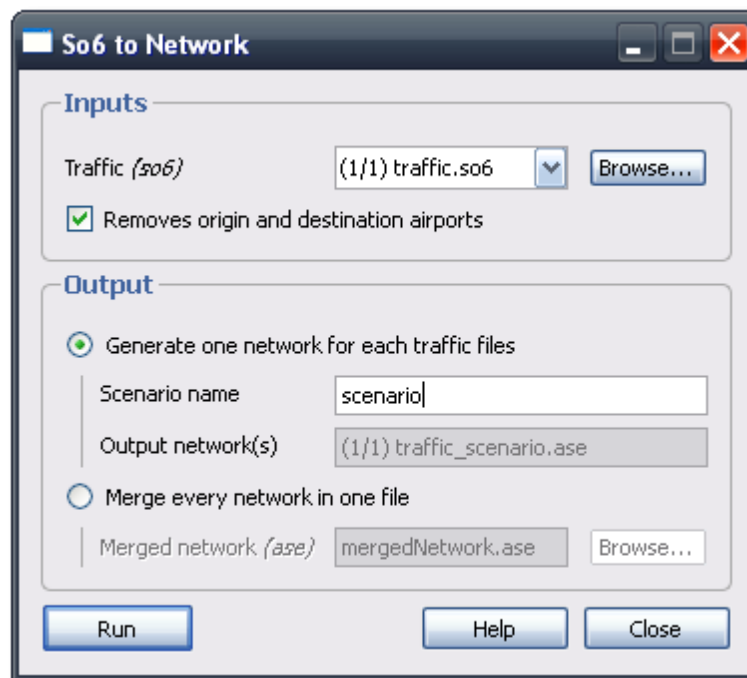
## Output

- Network file (\*.ase)

## General Procedure

1. Specify the input Traffic file
2. Specify the output format (multiple or merged)

3. Specify the scenario name (optional) or the name and location of the merged file.
4. Click Run



**So6 To Network Dialog Box**

## Parameters

1. **Traffic files field** – Browse one or several Traffic files (\*.so6)
2. **Removes origin and destination airports check-box** - If checked, the airports won't be added in the network.
3. **Generate one network for each traffic files check-box** - If checked, one network is generated for each input traffic files. The location and names of the files are display in the field **Output network(s)**.
4. **Merge every network in one file check-box** - If checked, all the network are merged in one file that you can specify in the field **Merged network**.

**Note:** The generated network file, output file, is a flown network. It will contain only route segments that at least one aircraft has flown. Number of flights (load) on each route segment is known i.e. a loaded network.

### 2.7.4 So6 to List of Options ( .lox )

#### Accessed by

[Menu Bar / Transform / So6 to ... / So6 -> List of Options \( .lox \)](#)

#### Purpose / Description

Converts the input traffic file (\*.so6) to a List of Options file (\*.lox). This process can be used for missing flights after an assignment. No additional route option is provided per flight, only

the original route will be present. This function should be used in the chain of processing aiming to provide full traffic load to [Optimisation](#) facility, also possibly to [Route Choice based on Military Opening Time](#).

## Input

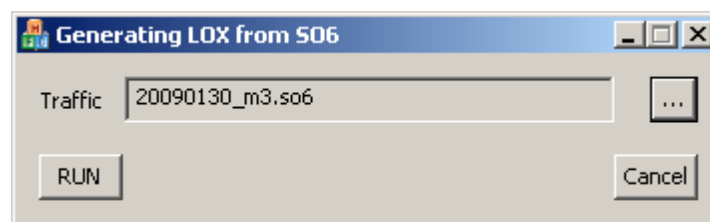
- Traffic file (\*.so6)

## Output

- List of Options file (\*.lox)

## General Procedure

1. Specify the input Traffic file
2. Click OK to Run



***So6 To List of Options ( .lox ) Dialog Box***

### 2.7.5 So6 to PIN

#### Accessed by

[Menu Bar / Transform /So6 to ... / So6 -> Pin](#)

#### Purpose / Description

Transforms the input so6 file into a pin file called by default with the name of the input so6 file with “\_saam.pin”, it can be changed. It is useful when SAAM users want to keep company routings as they were filed in CFMU flight plan but need to change only the profile constraints. This function can also work on a sub set of flights extracted from the [Query](#), on which profile processing will be launched, and the result will then be re-combined with other flights to get a new version of a “pseudo” current situation.

## Input

- Traffic file (\*.so6)

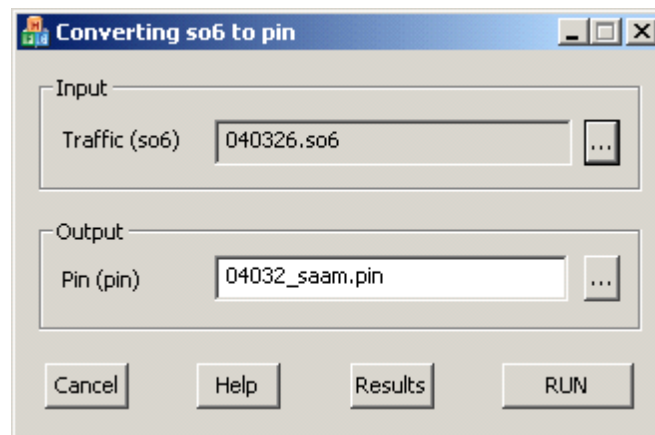
## Output

- Traffic file (\_saam.pin)



## General Procedure

1. Specify the input Traffic file
2. Click OK to Run



*So6 To Pin Dialog Box*

### 2.7.6 So6 To Direct Route PIN

#### Accessed by

[Menu Bar / Transform / So6 to ... / So6 -> Direct Rte PIN](#)

#### Purpose / Description

Takes one or several So6 files and changes them into Direct PIN. Which are used for profile calculation.

#### Input

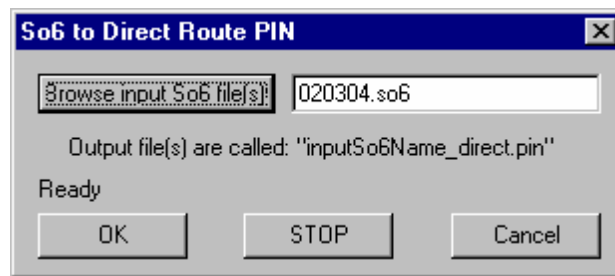
- Traffic file (\*.so6)

#### Output

- The output file name will be 'InputSo6Name\_direct.pin'

## General Procedure

1. Specify the input Traffic file
2. Click OK to Run



***So6 To Direct Route PIN Dialog Box***

## 2.7.7 S06 to Scenario

### Accessed by

**Menu Bar / Transform / So6 to ... / So6 to Scenario**

### Purpose / Description

Converts an existing traffic file (\*.so6) to a complete SAAM scenario, i.e. a set of files that can be edited.

### Input

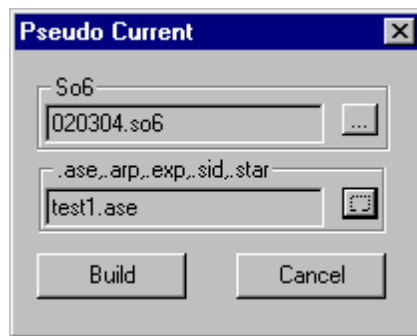
- Traffic file (\*.so6)

### Output

- Expand file (\*.exp)
- Network file (\*.ase)
- SID & STAR files
- Airport co-ordinate files (\*.arp)

### General Procedure

1. Specify the input Traffic file
2. Allocate a name to the output files
3. Click OK to Run



**So6 To Scenario Dialog Box**

## 2.7.8 Flow Demand to S06

### Accessed by

[Menu Bar / Transform / Flow Demand to So6](#)

### Purpose / Description

Takes a Flow Demand file and transform it to a So6 Traffic file.

### Input

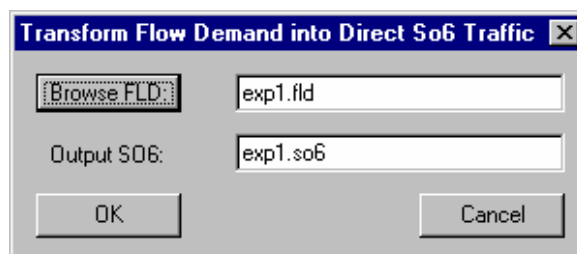
- Flow Demand file (\*.fld)

### Output

- Traffic file (\*.so6)

### General Procedure

1. Specify the input Flow Demand file
2. Allocate a name to the output file (*optional*)
3. Click OK to Run



**Flow Demand to So6 Dialog Box**

## 2.7.9 Matching Files

### Accessed by

[Menu Bar / Transform /Matching Files](#)

### Purpose / Description

'**Matching Files**' is used to extract lines of data from a data file when both key built from Key file and key built from data file are matching. The method is rather generic and can be used in many cases.

The key is a piece of information built from one or more field position(s), in any order.

For instance "2,1,3" means the key is built from 3 fields: field position 2 + field position 1 + field position 3. When the key built from the index file matched exactly the key built from the data file, the whole line from the data file is output. The maximum field position is 99, and a key can be composed of maximum 30 fields.

Note: if the key is built with more than one field, then the sub fields of the key are separated with "-". For instance: a key build with "2,1,3", with field 2 corresponding to a callsign, field 1 to airport departure and field 3 an aircraft type, internally the composed key will look like: "AFR1234-EGLL-A319".

### Input

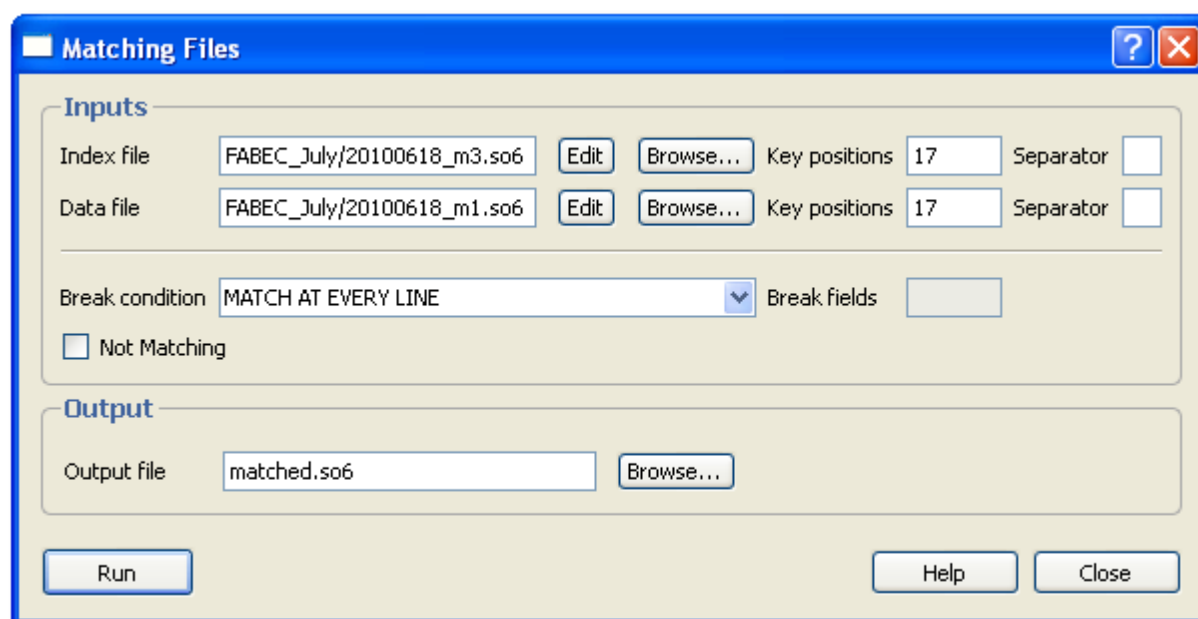
- An Index file, key field(s) position(s), field separator in key file
- A Data file, data field(s) position(s), field separator in data file

### Output

- Output file will have the same format as the Data file. The dialog box is adding, if necessary, the correct extension to the output file.

### General Procedure

1. Specify the Index file
2. Specify the field position (s) to built the key from the index file (field 1 means first column)
3. Indicate the field separator used in the index file (\t means tab separated)
4. Specify the Data file
5. Specify the field position(s) to built key from the data file (field 1 means first column)
6. Indicate the field separator used in the data file (\t means tab separated)
7. Chose a '**Break Condition**' option. If 'MATCH AT BREAK LINE' is chosen you have to specify 'Break Field'.
8. Allocate a name to the output file (warning: it must have the same extension than the Data file!!)
9. Click '**RUN**' to start the process.



**Matching Data Files Dialog Box**

## Parameters

1. **Index File *field*** - The template file that will provide the key is called the 'index' file.
2. **Index File Key Positions *field*** - The user must specify the position of the 'key' field/s. Where multiple keys are used their numbers will be separated by commas without space.
3. **Index File Separator *field*** - The separation between fields is by default a space. If it is a TAB, it has to be written as '/t'. If input file is \*.exp or .so6 file than separator is set to a space, if it is an \*.exp2 file it is set as a semi column ";".
4. **Data File *field*** - The file which contains the data to search. It can be of a different type than the Index file.
5. **Data File Key Positions *field*** - To indicate the position of the 'key' field/s. Where multiple keys are used their numbers will be separated by commas. Same comment as the one stated in 2.
6. **Data File Separator *field*** - The separation between fields is by default an empty space. If it is a TAB, it has to be written as '/t'. Same comment as stated in 3.
7. **Edit buttons** - The accuracy of the operation depends on the precise identification of the key(s) and it's (their) position. Therefore a viewing function has been featured in order to examine the structure of the files (the head of the files is viewed).
8. **Break condition drop-down list** - Indicates the break detection mode. There is two possible values:
  - a. **MATCH AT EVERY LINE :**  
This is the default behavior. Every lines are processed one by one.
  - b. **MATCH AT BREAK LINE :**  
This mode can be used when only the first line of a group of few consecutive line shall be processed. It is useful, for instance, to retrieve the first segment of each flight of an SO6 file.
9. **Break fields *field*** - Only available if the break condition is set to "MATCH AT BREAK LINE". The format is the same as the Key Positions fields. Every consecutive lines with

identical the break fields are considered as one group. Only the first line of such a group is processed.

Example

<b>Key1</b>	<b>field1</b>	<b>field2</b>	<b>field3</b>	<b>brekField1</b>
<b>-&gt; Processed</b>				
<b>Key2</b>	<b>field4</b>	<b>field5</b>	<b>field6</b>	<b>brekField1</b>
<b>Key3</b>	<b>field7</b>	<b>field8</b>	<b>field9</b>	<b>brekField2</b>
<b>-&gt; Processed</b>				
<b>Key4</b>	<b>field10</b>	<b>field11</b>	<b>field12</b>	<b>brekField2</b>
<b>Key5</b>	<b>field13</b>	<b>field14</b>	<b>field15</b>	<b>brekField2</b>

10. **Not Matching check-box** - It is possible to toggle between 'matching' (the data which will be extracted matches the key) and 'not matching' (the data which is extracted doesn't match the key; the matching data is disregarded)

## Fields position(s) auto-filling

To facilitate usage of matching the following data files are recognised: exp, exp2, so6, allft, and depending on which is key and which is data the field(s) position(s) and separator are proposed automatically (it can be modified by user).

The following table indicates the proposals made by the dialog box for the field(s) position(s):

		data		
		exp2	so6	ALLFT
key	exp2	2 <=> 8	8 <=> 17	12,1,2,42 <=> 3,1,2,8
	so6	17 <=> 8	17 <=> 17	17 <=> 23
	ALLFT	3,1,2,8 <=> 12,1,2,42	23 <=> 17	23 <=> 23

You will notice that ALLFT and exp2 keys are built on: callsign, ADEP, ADES, IOBT, while universal key present in field #15 in exp2 is build on callsign, ADEP, ADES, EOBT. Currently we could not use universal key as its formatting was not finalized and EOBT (field #19 in ALLFT and field #52 in exp2) is currently not always the same. When data files fields values will be OK then matching could be done on field #15 or composed using EOBT instead of IOBT.

Matching between so6 and exp2 or so6 and ALLFT, might not always work. Fields #8, fields #17 and field #23 contains flight ID. These flight ID might not have the same origin: PRISME ID or TACT ID or other, depending on how files were build (for instance translated from another source) and where the files are coming from (DDR, EEC, CFMU ...).

## Examples:

### Example 1 - Add information to a file

Consider a key file consisting of Aircraft type, Origin, Destination and Color of seats

**B733 EBBR EGLL R**

**A380 LFPG KLAX B**

**DC3 GABS DFFD Y**

**A330 LTBA LGAV G**

and a data file consisting of Origin Destination RFL Callsign Aircraft type.

**LFPG KLAX 370 AFR200 A380**

**EBBR EGLL 230 SAB666 B733**

**EBBR EGLL 250 BAW200 B732**

**LTBA LGAV 330 AZA537 A330**

The key file position should be: 2,3,1 (the key is made of airport origin + airport destination + aircraft type) and the data file position should be: 1,2,5 (the key is made of airport origin + airport destination + aircraft type, the same as the one in key file). By matching this data (continuous data type, matching data). The result will be (extract of lines coming out from data file that are matching the key file):

**EBBR EGLL 230 SAB666 B733**

**LFPG KLAX 370 AFR200 A380**

**LTBA LGAV 330 AZA537 A330**

Line 3 of the key file is disregarded as it is not in the data file. Line 3 of data file is also disregarded as the matching aircraft type is a B733 (and not a B732 !). The 4th field in the key file is also disregarded.

### **Example 2 - Retrieving so6 data which has been lost after assigning traffic**

While assigning traffic on a Vx network, all dubious flights are eliminated from the resulting so6 file. In order to obtain a complete sample those flights can be retrieved from the current network sample.

The key file is the \*\_missing.exp or \*\_missing.exp2 file coming out from assignment. The key position is field #8 (flight number identification, automatically selected). The data file is the original CFMU current traffic sample of the same day. The key position is field #17 (flight number identification, automatically selected). Use the 'MATCH AT EVERY LINE' option (as the data is continuous) and check the 'Matching' option. Check that the output file as the same file extension than the input data file (because it is an extract of it and then it has the same type).

The output contains the flights that have not been assigned routed on CFMU flight plan data. This file can be re-injected using menu Transform / Combine SO6, T5, LOX.

### **Example 3 - Excluding Helicopters from EXP(2) files**

In macroscopic analyses, IFR flights operated with helicopters are often not of interest for a study. The text below describes how to exclude helicopters from an EXP(2) file using this "matching" process.

The key file is "helicopters.txt" which you will get from "[Menu Import/Get Basic Airspace Data/Helicopter list](#)". The key position is 1. In field for the Data File, enter the EXP or EXP2 file from which the flights operated with helicopters have to be removed. Select Data File Position 4 (for both EXP or EXP2 files).

Set the Break Condition to "Match at every line" and enter a new name for the Output File (by default, this will be "matched.exp"). Also select the box "Not Matching", as this will actually exclude the matching lines from the original file.

## 2.7.10 Lowering S06 FL

### Accessed by

[Menu Bar / Transform / Lower So6 FL](#)

### Purpose / Description

Recent developments in ATM (RVSM, 8.33) have reserved the upper airspace to recent and equipped aircraft. Thus non-compliant aircraft have to fly lower than they used to. This adjustment can be taken into account using the '**Lowering FL**' option. This is done by associating a so6 file with a text file that contains the new FL. The output will be a new so6 file. The only thing which is modified is the max. FL, thus the flat route and the flight times are unchanged.

### Input

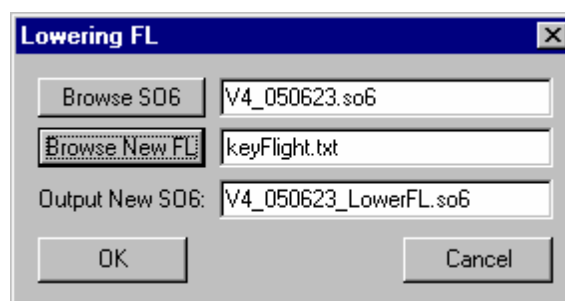
- Traffic file (\*.so6)
- Text file (\*.txt) containing new FL

### Output

- Traffic file (\*.so6)

### General Procedure

1. Specify input traffic file (\*.so6)
2. Specify input FL file (\*.txt)
3. Assign a name to the output Traffic file
4. Click OK to Run



**Lowering FL Dialog Box**



## Parameters

The format of the text file is:

**#of flight new flight level ... optional additional info ...**

**8469 280 IL76**

**5142 270**

**5141 400**

**19036 260**

This function has an idiot-proof capability, i.e. mistakes in the input data are disregarded but stored in lowering\_error.txt or lowering\_warning.txt.

Most common type of errors are: duplicated flight # or unknown flight # (line 4 in our example).

Flights with unsuitable FL, i.e. with a new FL which is too high (line 3 in our example) are stored in the lowering\_warning.txt.

## 2.7.11 Airspace to TDV

### Accessed by

[Menu Bar / Transform / Airspace to TDV](#)

### Purpose / Description

This function is used to create a TDV of an airspace (sectors, volumes, etc.). For that purpose it creates individual sectors shapes (a file per sector) in the directory "ARE" created in your working directory. This last operation is also used in [Sector Workload](#) (macroscopic formula) to generate a overloaded sector map, and with ColorTDV2.exe from [Generic Launch](#).

Note: it is not more required by [Query](#) !

### Input

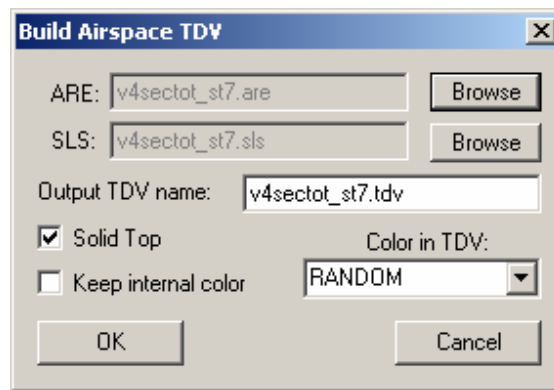
- Airspace file (\*.are)
- An \*.sls file, which is a description of the way the basic volumes are combined in order to build more elaborate pieces of airspace.

### Output

- Three Dimensions View file (\*.tdv)

### General Procedure

1. Browse input airspace file (\*.are)
2. Browse input (\*.sls) file
3. Assign a name to the output Tdv file
4. Click OK to Run



**Build Airspace Tdv Dialog Box**

## 2.7.12 Traffic Punching

### Accessed by

[Menu Bar / Transform / Traffic Punching](#)

### Purpose / Description

For simulation and animation purposes, it is necessary to extract the traffic which crosses an airspace. **Traffic Punching** gives you several methods to do so.

### Input

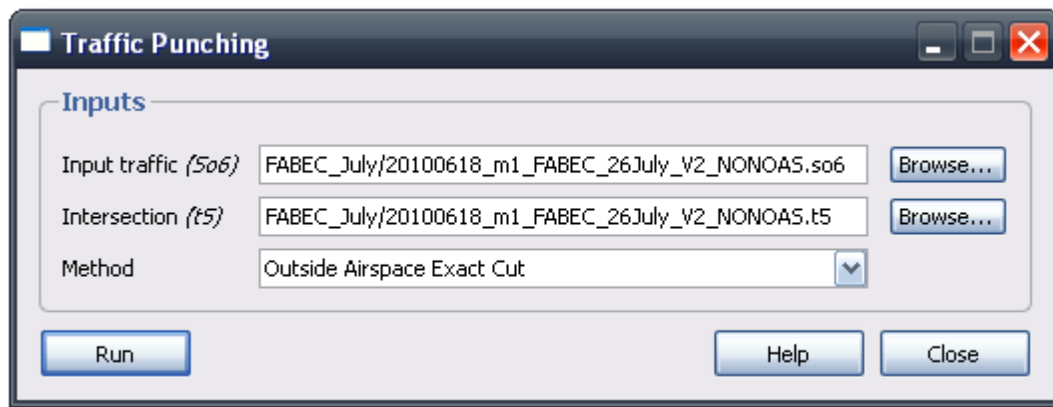
- Traffic file (\*.so6)
- Traffic Intersection file (\*.t5)

### Output

- Traffic files equals names of airspace found in the t5 file

### General Procedure

1. Specify input Traffic Intersection file (\*.t5)
2. Specify Traffic file (\*.so6)
3. Chose a method
4. Click OK to Run



**Traffic Punching Dialog Box**

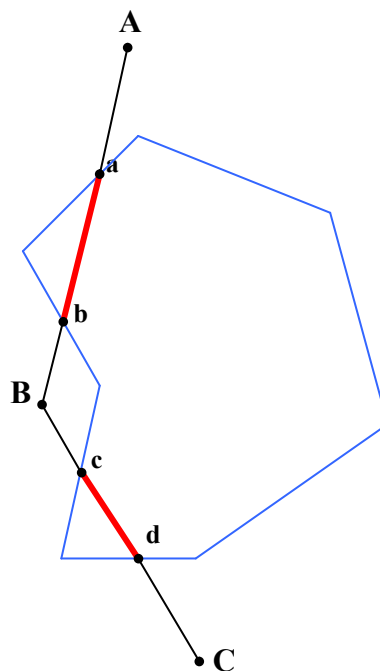
## Parameters

Several methods can be used to extract traffic which crosses an airspace.

If a portion of the route crosses the selected airspace the whole route is taken into account. This is called traffic extraction (See supra in the [Query box](#)).

If you want to extract just a part of the route you can use one of these methods.

- **Method** drop-down list –
  - a. **Inside Airspace Exact Cut** - This process only takes into account the portions of segments that are comprised in the selected area (thus in our example portions a-b and c-d).
  - b. **Outside Airspace Exact Cut** - To select the traffic outside the punching area using the same criteria as above.
  - c. **Inside Airspace First In Last Out Segment Cut** - The segments that have a portion which is comprised within the airspace are taken into account (Thus in our example A-B and B-C).
  - d. **Outside Airspace First In Last Out Segment Cut** – To select traffic outside the punching area using the same criteria as above.

**Example:****2.7.13 Conflict Traffic Cut to Animation****Accessed by**

**Menu Bar / Transform / Conflict Traffic Cut to Animation**

**Purpose / Description**

Generates a TDV file showing an animated traffic cut within the area where [Conflict](#) was processed. It shows an aircraft and all other aircrafts with which conflicts were detected.

**Note:** A SIM diagram which run the Conflict and this process with default parameters is available within SAAM.

**Input**

- Conflict file (\*.conf) produced for one flight ! See the single flight filtering of the conflict dialog.
- Traffic file (\*.so6) should also containing the flight for which the conflict were processed (which is normally the default case).
- A Flight ID number. It has to be the same as the one specified when the conflict calculation was done.
- The radius of the cylinder which will be draw over the flight in the animation. It is usually the same value as the horizontal separation distance specified in the conflict dialog box.

**Output**

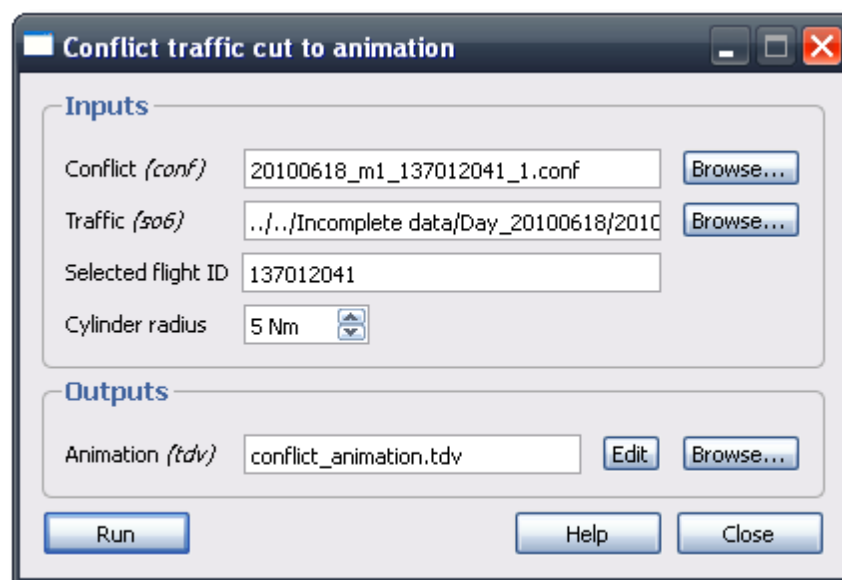
- A TDV file (\*.tdv). It contains the flight for which the conflicts were calculated. The flight

is animated as a blue "B737" aircraft shape, and as yellow transparent limited cylinder. The TDV also contain animated aircraft, displayed in red, corresponding to the "\_cut.so6" file (see line below).

- One so6 traffic files called with the name of ouput TDV with suffixe "\_cut.so6". This file contains all traffic conflicting more than 0 second ("touching" conflict are eliminated because it would be impossible to make an animation or to display the trajectory) with trajectories cut within the horizontal & vertical separation area defined during conflict calculation. This file is animated in the generated TDV. This file is stored in a folder called <tdv\_name>\_data next to the tdv file.
- One so6 traffic file called with the name of ouput TDV with suffixe "\_full.so6". This file contains all traffic conflicting more than 0 second ("touching" conflict are eliminated because it would be impossible to make an animation or to display the trajectory) with full trajectories from origin to destination. This file is stored in a folder called <tdv\_name>\_data next to the tdv file.

## General Procedure

1. Specify Conflict file (\*.conf)
2. Specify Traffic file (\*.so6)
3. Select a Flight ID number (mandatory).
4. Select the appropriate cylinder radius.
5. Type a name to the output TDV file
6. Click OK to Run
7. Open the generated TDV, possibly add trajectories with queries.



**Conflict traffic cut to animation dialog box**

## 2.7.14 Combine SO6, ARE or LOP

### Accessed by

[Menu Bar / Transform / Combine SO6, ARE or LOX](#)

### Purpose / Description

Merges two or more files (max 7) into one.

### Input

Two or more (max 7)

- Traffic files (\*.so6 and \*.t5) or
- Airspace files (\*.are and \*.sls) or
- List of Options files (\*.lox) or
- Network file (\*.ase)

### Output

one output file:

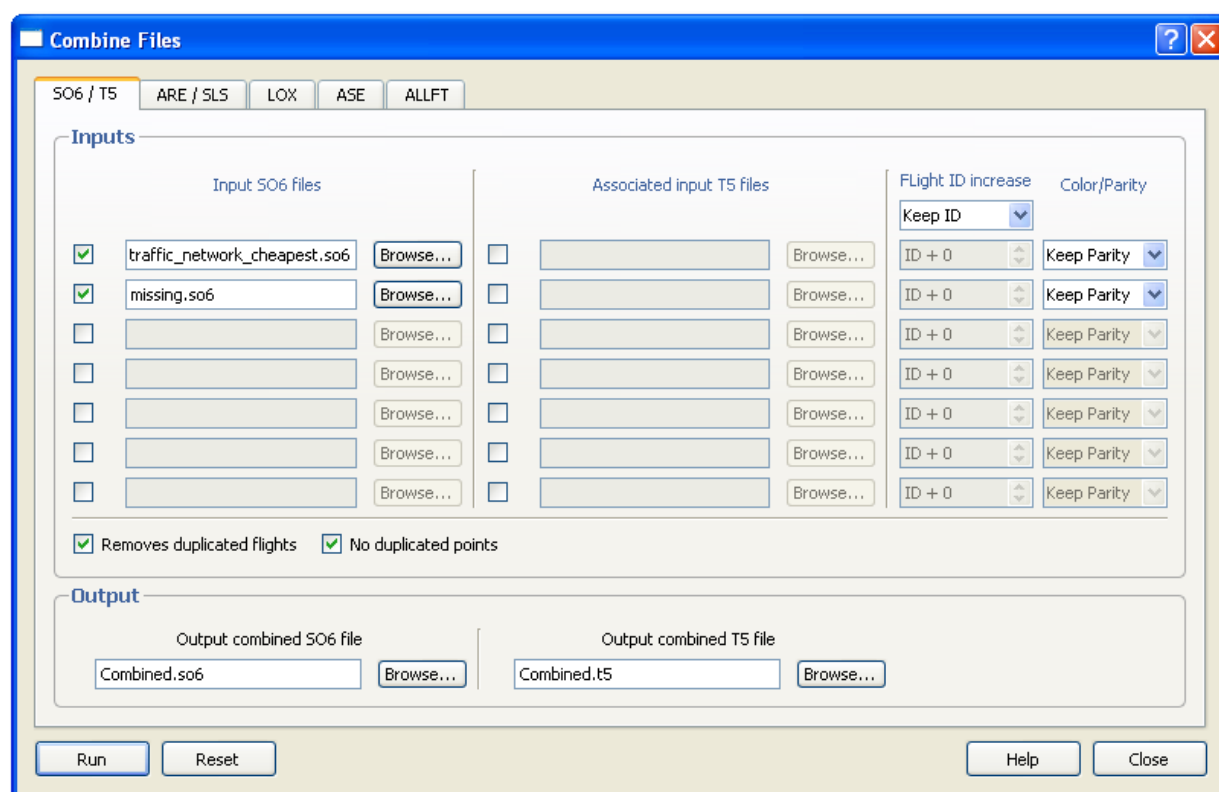
- Traffic file (\*.so6 and/or \*.t5 ) or
- Airspace file (\*.are and/or \*.sls) or
- List of Options file (\*.lox) or
- Network file (\*.ase).

### Note

No care is provided to insure that input files to combine are not sharing same information, or even information having the same name but being different. For instance, combining networks that have different coordinates for same point name or segment is not checked !! So combining networks function is useful when networks are either geographically different or have different segments sharing same point coordinates (like combining a network of DCT segments on top of existing route network).

### General Procedure

1. Select the tab with the type of file you want to combine.
2. Specify input files
3. Optionally change the options of the selected tab
4. Choose a name to the output files (*optional*)
5. Click Run



**Combine Files Dialog Box**

## Parameters

1. **SO6/T5, ARE/SLS, LOX, ASE and ALLFT tabs** – Select the tab corresponding of the type of file you want to combine. You can combine only one type of file at a time.
2. **Browse Input files drop-down list** – Check the check-box to activate the Browse button and select the files to be combined. Either a Traffic files (\*.so6), an Airspace files (\*.are), a List of Options files (\*.lox), or a Network files (\*.ase) depending of the selected tab.
3. **Browse associated input files drop-down list** – Check the check-box to activate the Browse button and select the associated files to be combined. If both an input file and an associated input file are activated on the same line, selecting one of them will automatically select the other one if it exist. You can also browse yourself the files. Only available for So6/T5 and Are/Sls tabs
4. **Flight ID increase numeric value** – Select a value to be added to the Flight ID. This is to make sure that every flight in the output file have a unique Flight ID. This option is generally not required and can be let as "Keep ID". Only available for the SO6/T5 tab.
5. **Colour/Parity drop-down list** – To make it possible to differentiate flights from one input file from another when visualised. Only available for the SO6/T5 tab.
6. **Removes duplicated flights checkbox** - If checked, flights with a previously defined flight ID will be removed from the output. Only available for the SO6/T5 tab.
7. **No duplicated points checkbox** - If checked, the process will ensure that two points with the same name (but with different coordinates) doesn't exist in the output. Duplicated points will be renamed with a random value. This is used to solve CFMU problem where same points could be reused with different coordinates at different days. Only available for the SO6/T5 tab.
8. **Output combined file field** – By default output files are called 'combined.\*' and is located in the scenario directory. You can changed them by browsing a new location.

9. **Run** button – Runs the process for every type of files.
10. **Reset** button – Resets all the inputs, outputs and parameters to their default values.

## 2.7.15 Sector\_SLS to Center\_SLS

### Accessed by

[Menu Bar / Transform / Sector\\_SLS to Center\\_SLS](#)

### Purpose / Description

Cuts ending characters of sector names found in input sls to transform them into a center. It's valid only if the sector name is build with classical rules; ie sector 'LFEEHX' will become center 'LFEE'.

### Input

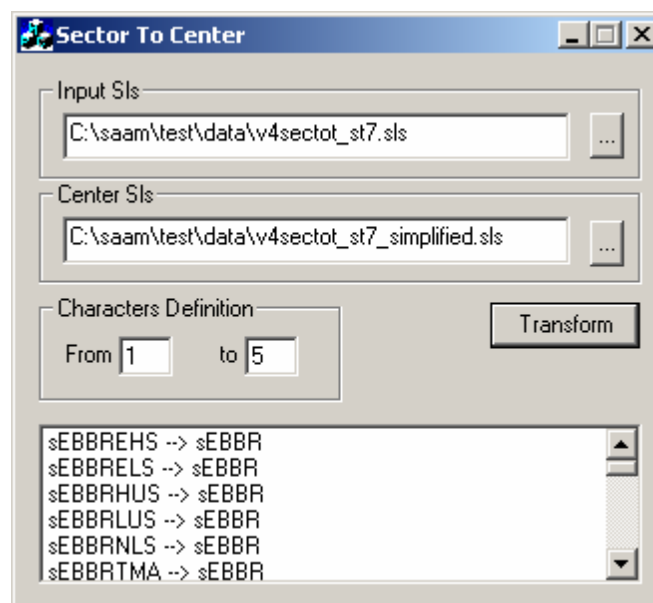
- Airspace files (\*.sls)

### Output

- Airspace file (\*.sls)

### General Procedure

1. Specify input files
2. Set the Characters Definition
3. Click Transform to Run



**Sector to Center Dialog Box**



## Parameters

1. **Input Sls** *field* – Browse a sls Airspace file
2. **Center Sls** *field* – Will automatically create a '**Center Sls**' file with the same name as the '**Input Sls**' file plus adding simplified to the name.
3. **Characters Definition** *fields* – An option to specify number of characters in the **Center** name.
4. **Transform** *button* - Starts the process.

## 2.7.16 Airspace 3D Envelop

### Accessed by

[Menu Bar / Transform / Airspace 3D Envelop](#)

### Purpose / Description

Takes an airspace as input (\*.are and \*.sls files) and process, slice per slice, the external envelop. The output is a new airspace files, \*.are and \*.sls, called like the input files with a suffix. The output files are located in current working directory. This function can be used for checking consistencies of vertices, holes and overlaps before exporting to RAMS, TAAM or IPAS. It can also be used to simplify airspace shapes by suppressing inner boundaries.

### Input

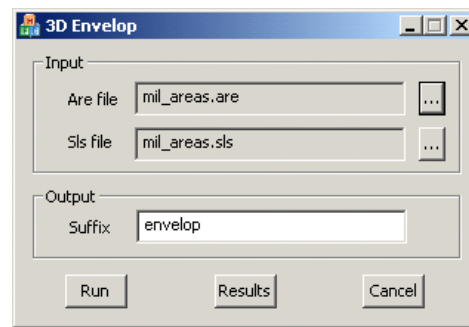
- Airspace file (\*.are)
- Airspace file (\*.sls)

### Output

- Airspace file\_ suffix (\*.are)
- Airspace file\_ suffix (\*.sls)

### General Procedure

1. Specify input files
2. Type a suffix
3. Click Run



**Sector to Center Dialog Box**

## Parameters

1. **Input frame** – Browse Airspace files, \*.are and \*.sls
2. **Output frame** – Adds a suffix to the output name. "envelope" is suggested, but can be changed

### 2.7.17 Extract Traffic within Time Period

#### Accessed by

[Menu Bar / Transform / Extract Traffic within Time Period](#)

#### Purpose / Description

Extracts traffic flights, from the input (possibly multiple) .so6 traffic file(s), that are present in any location within the areas given as input (intersection is handled internally) during the specified time period.

Output is one traffic file without duplication of flights.

Case of midnight is handled allowing to extract night traffic (for instance from, let's say 22h00 till 03h00). Entire 4D flight trajectories are always extracted.

The time criteria is NOT the entering time, it is the presence within the area during the time period which is used, as consequence an animation of the resulting extracted traffic will probably show aircraft flying within the area before or after the time period. but the flight will in any case be somewhere within the area during the time period.

If several input traffic are given and if a night traffic is requested, the "morning" of the first day and the "evening" of the last day are not extracted by default. But this behaviour can be changed from the dialog box. Note: if a single traffic day is provided in the input, and if a night extraction is requested, then morning and evening periods of that day are always extracted.

The instantaneous number of flights curve of the extracted traffic compared to the original traffic are the same for the time period. The SHER curve will be identical for the time period if the beginning of the period starts one hour before (because by definition the SHER works for the following hour).

## Input

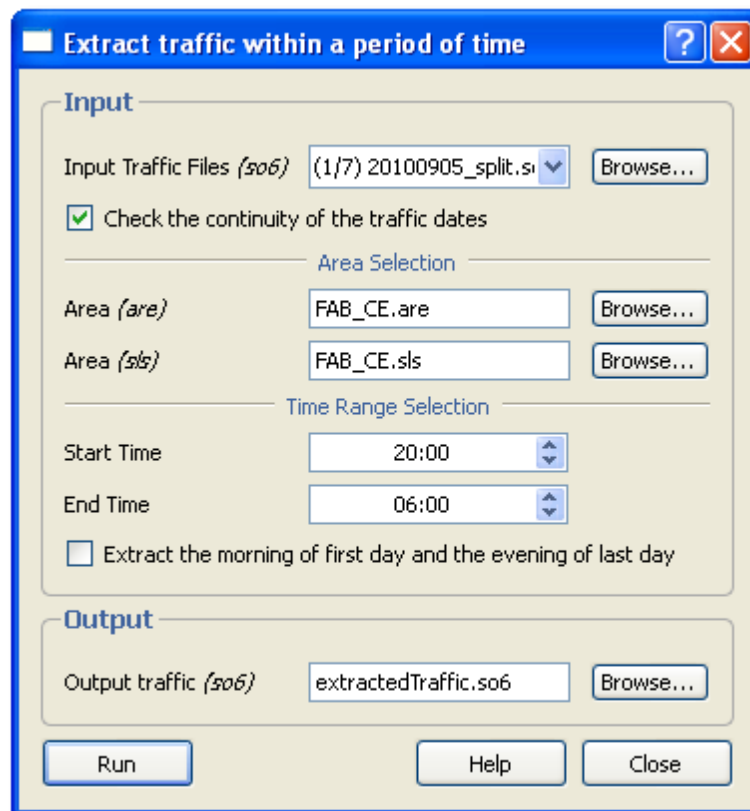
- Traffic file (\*.so6)
- Area used to process intersection (integrated) (\*.are/\*.sls)
- Start time and end Time (must be swapped for a night time period).

## Output

- One Traffic file (\*.so6)

## General Procedure

1. Browse one or several traffic file
2. Browse the area
3. Set the start and end time (swap values for night traffic)
4. Type proper name for output so6 file
5. Click on Run.



**Extract Traffic within Time Period Dialog Box**

## Parameters

1. **Traffic (so6) field** – Browse a traffic file (\*.so6)
2. **Check the continuity of the traffic dates checkbox** - If checked, the continuity of the input traffic file is verified.

3. **Area (are/sls) field** – Browse area file (\*.are/\*.sls)
4. **Start / End time fields** – Set the start and end time for which traffic will be extracted (swap values for night traffic)
5. **Extract the morning of the first day and the evening of the last day checkbox** - This option is only available if the midnight barrier is crossed (i.e. start time is greater than end time). Note: if a single traffic day is provided in the input, and if a night extraction is requested, then morning and evening periods of that day are extracted regardless of this option.
6. **Name for the output file field** – Browse and type a new name for the output file.

## Example

### How to extract a 3 night traffic from 10pm to 4am in a given zone ?

A proper extraction will require 4 successive days (to be taken from DDR for example). Browse your area (must be pre-defined or draw it with Airspace Editor). Type the time: start time will be 22:00, end time will be 03:59. Browse and type your name for the output so6 file. Launch it.

## 2.7.18 Network Extraction

### Accessed by

[Menu Bar / Transform / Network Extraction](#)

### Purpose / Description

To extract a part of a route network lying inside or outside of a given area. The extraction methods, inside or strictly outside, are complementary. Meaning so that route networks being extracted with these two methods can be merged again.

### Input

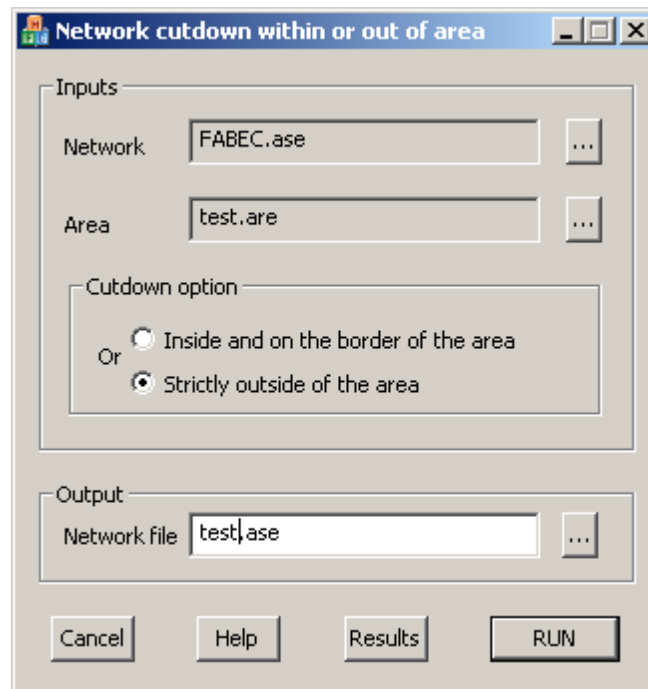
- Route network file (\*.ase)
- Airspace file (\*.are)

### Output

- part of a route network (\*.ase)

### General Procedure

1. Browse the route network file
2. Browse the airspace file
3. Select method -           Inside and on the border of the area  
                                     Strictly outside of the area
4. Give a name to the output network file
5. Click on Run.



**Network Extraction Dialog Box**

## 2.7.19 Data file decompression

### Accessed by

[Menu Bar / Transform / Data file decompression](#)

### Purpose / Description

A facility to dempress data files. It can decompress one or several files, and delete them once decompressed. It works with 7za decompressor and accept .zip, .gz and .rar., but not .zipx.

### Input

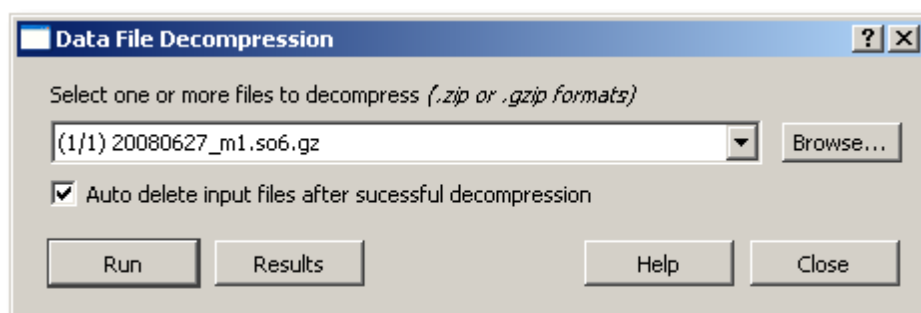
- One or several decompressed files (\*.zip, \*.gz or \*.rar)

### Output

- Un-compressed files

### General Procedure

1. Browse one or several decompressed files
2. Select auto delete (optional)
3. Click on Run.



*Data file decompression Dialog Box*

## 2.7.20 Network Coordinate Check

### Accessed by

[Menu Bar / Transform / Network Coordinates Check](#)

### Purpose / Description

Checks the names and the positions of the points in a network file (\*.ase) using a list of reference points extracted from the DDR (\*.navpoint).

### Input

- One or several network (\*.ase)
- The list of reference points from the DDR (\*.navpoint)

### Output

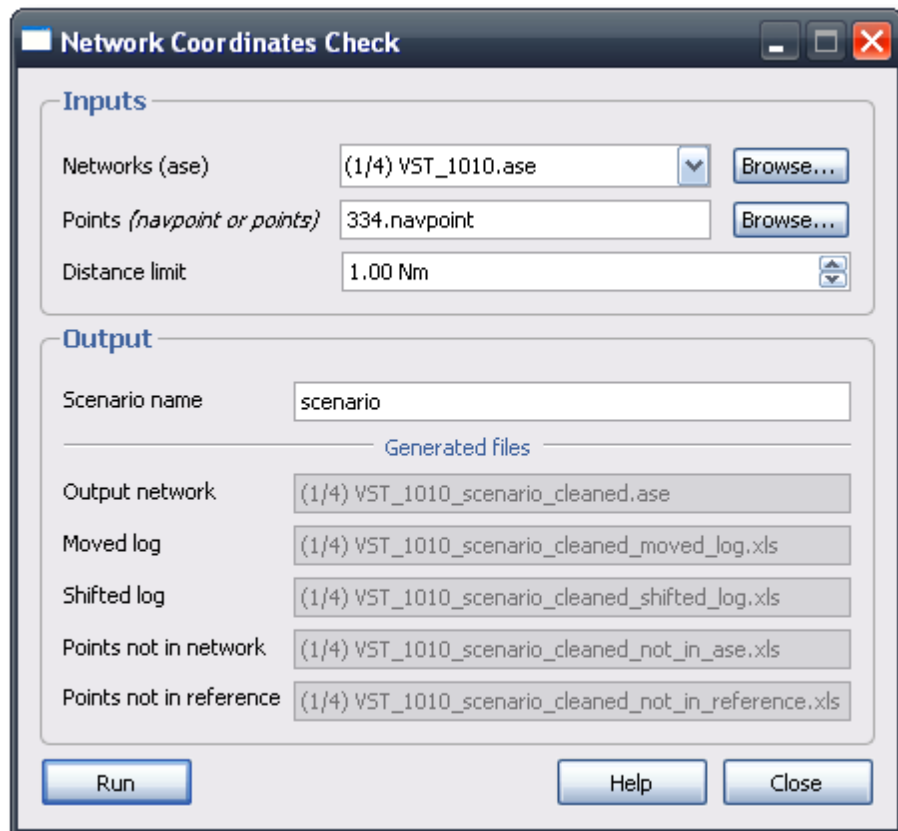
For each input file, five output are generated:

- **Output network** (\*\_cleaned.ase): The input network which has been cleaned. Every point, that has a difference of position its reference lesser than the distance limit is automatically moved at its correct position.
- **Move log** (\*\_cleaned\_moved\_log.xls): The list of point that have been moved.
- **Shifted log** (\*\_cleaned\_shifted\_log.xls): The list of point that has not been moved and that stay shifted from their original position.
- **Points not in network** (\*\_cleaned\_not\_in\_ase.xls): The list of point which exist in the reference points but not in the network.
- **Points not in reference points** (\*\_cleaned\_not\_in\_reference.xls): The list of point which exist in the network but not in the reference points.

### General Procedure

1. Specify the input network files
2. Specify the point file
3. Change the distance limit if needed
4. Specify the scenario name (optional)

## 5. Click Run



**The Network Coordinates Check Dialog Box**

## Parameters

1. **Networks field** – Browse one or several network files (\*.ase)
2. **Points field** - Browse the point file
3. **Distance limit value** -The limit which determine if the position a point in the input network is correct or incorrect.
4. **Scenario name field** - Specify the suffix to be added to the output file names. Optional.

## 2.8 Pick Info Menu

Not yet implemented

## 2.9 Export Menu

### Accessed by

[Menu Bar / Export](#)

### Purpose / Description

Converts SAAM data into other file formats used by other applications.

### 2.9.1 TAAM

### Accessed by

[Menu Bar / Export / TAAM](#)

### Purpose / Description

Exports SAAM files (traffic and/or sectors) into TAAM files: .acf, .wpt, .rts. The traffic can be cut within a window or not. The profile constraints are not yet exported.

### Input

- Traffic file (\*.so6)
- Cut airspace file (\*.are) *Optional*
- Airspace files (\*.are and \*.sls) *Optional*

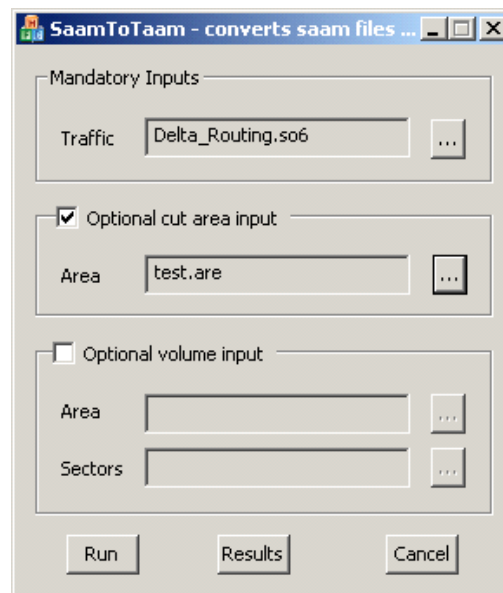
### Output

- .acf
- .wpt
- .rts

### General Procedure

1. Browse traffic file (\*.so6)
2. Browse cut airspace file (window) (\*.are)
3. Browse airspace files (\*.are and \*.sls)
4. Click on Run





**Dialog Box**

## 2.9.2 IPAS

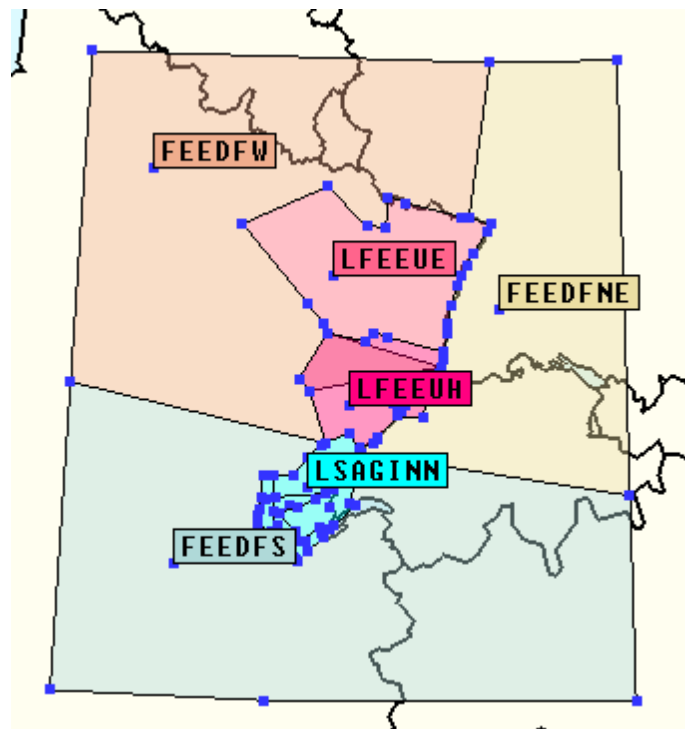
### Accessed by

[Menu Bar / Export / IPAS](#)

### Purpose / Description

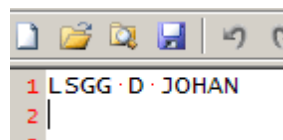
Exports SAAM files (traffic, sectors and profile constraints) to EEC / Budapest Real Time simulator via the IPAS interface. It produces, in the user specified specific directory (by default "IPAS\_files") created in the working directory, a number of \*.csv files, plus an so6 and a t5 file. This process extract traffic crossing measured sectors within specified time period, and cut trajectories in a way that they start/finish in the feeder sectors. Important: the constraint for IPAS are strong. Airspace file, provided by the user, must contain feeder sectors around, above and below measured sectors to be simulated (to cover airspace from ground to FL600), no holes and no overlap are allowed (see Airspace 3D Envelop function). Feeder sectors, as well as measured sectors, can be built with several airblocks. Name of sectors (present in first column of .sls file) must be uppercase (without strange characters) and must have the following size: ACC\_name (4 characters, must be "FEED" for feeders sectors) + real sector name (max 5 characters).

The shape of feeder sectors must be aligned in the north and south to parallels and on the west and east to meridians (use copy/paste of points coordinates, either latitude or longitude, while right clicking on feeders vertices). It is possible to have vertices at other places than corners, but they must be aligned as specified above.



**Top view at a given DFL of measured and feeder sectors**

Optional management of suppression of SID/STAR points to/from airports within the area to be simulated is triggered via a user input text file having .ssl extension (if empty the SID/STAR management does not take place, but the conversion to IPAS will be valid). This SID/STAR \*.ssl file contains 3 columns: airport\_name code\_A\_D(A or D for Arrival or Departure) point\_name (=last point of the SID or first point of the STAR). Possible errors are output in file "SID\_STAR\_errors.txt". Once intermediate points are suppressed (so excluding points present in the SID/STAR file) they can be replaced by IPAS team preparing RT simulation with the ad hoc SID/STAR definition (outside SAAM scope).



**Departure points list from Geneva till (excluding) JOHAN will be cleaned**

One output file is a sector entry list. It gives for each flight the ordered list of penetrated sectors, including entry time and entry level.

The traffic \*.so6 and \*.t5 files that are generated with IPAS files corresponds to the traffic that was extracted and cut. It can be used with a Query to check and validate with airspace designers the routings, levels, sectors sequence of the traffic to be simulated.

## Input

- Traffic file (\*.so6)
- Airspace files (\*.are and \*.sls)
- Flight Level Constraint file (\*.flc2)
- Optional SID/STAR text file (\*.ssl)

- output directory name (default is "IPAS\_files")

## Output

Several \*.csv files found in directory "IPAS\_files"

- sectors definition
- sectors points
- traffic definition
- traffic nav aids
- traffic \*.so6 and \*.t5 file(only for checking with SAAM)
- airways (for IPAS display)
- flight level constraints
- Sector entry list

## General Procedure

1. Specify input files
2. Set the Start and End time
3. Select the traffic increase
4. Select an output directory
5. Click OK to Run



**SAAM to IPAS Dialog Box**

## Parameters

1. **Traffic (so6) field** – Browse
2. **Sector (are) field** – Browse
3. **Sector (sls) field** – Browse
4. **Constraint (flc2) field** – Browse a constraint file. This file could empty, but must contain required header and footer.
5. **SID/STAR list (ssl) field** - This is optional can be empty (no space).
6. **Start / End time field** – Set the start and the end time for cutting the traffic within the sectors.
7. **Traffic increase fields** – An option to increase the traffic with a specified percentage.
8. **Maximum time deviation in minutes, by step of 5 field** – To increase traffic, flights will be duplicated. Only the departure time will be changed. Here you can set the maximum change in departure time, in minutes, for duplicated flights.
9. **Output directory** - Select the directory where all the IPAS files will be created.

### 2.9.3 RAMS

#### Accessed by

[Menu Bar](#) / [Export Menu](#) / [RAMS](#)

#### Purpose / Description

Converts SAAM traffic with possibly FL constraint (LOA in the form of flc2 file), and/or Airspace (Sectors) with possibly configuration (.cfg file) data into RAMS file format. You will notice that user can convert either Traffic, or Airspace or both.

#### Input

- 4D trajectories (\*.so6)
- Flight Level Constraints (\*.flc2) is optional.
- Traffic Selection area (\*.are and \*.sls, \*.gar and \*.gsl)
- Traffic Cut Area (\*.are and \*.sls, \*.gar and \*.gsl)
- Airspace files (\*.are and \*.sls, \*.gar and \*.gsl)
- Sector configuration (\*.cfg) is optional.

#### Output

- set of RAMS airspace data files stored in a directory (to be created or browsed).

#### General Procedure

1. Specify input files for the traffic conversion: input traffic, traffic selection and cut areas (these selection and cut areas must exist or be created with SAAM Airspace Editor before launching the conversion)
2. Specify the input files for airspace conversion.
3. Type the output file name directory (all RAMS files will be created in this directory)
4. Enter a scenario name
5. Check the "Copy RAMS default files" checkbox if necessary.

6. Click Run.

**SAAM to RAMS**

☒ **Input Traffic**

Traffic Trajectory (*so6*)

Flight Level Constraints (*flc2*)

**Traffic Selection**

Inside Area (*are or gar*)

Inside Area (*sls or gsl*)

**Traffic Cut**

Outside Area (*are or gar*)

Outside Area (*sls or gsl*)

☒ **Input Airspace**

Sectors (*are or gar*)

Sectors (*sls or gsl*)

☐ Sector configurations (*cfg*)

**Output**

RAMS Data Directory

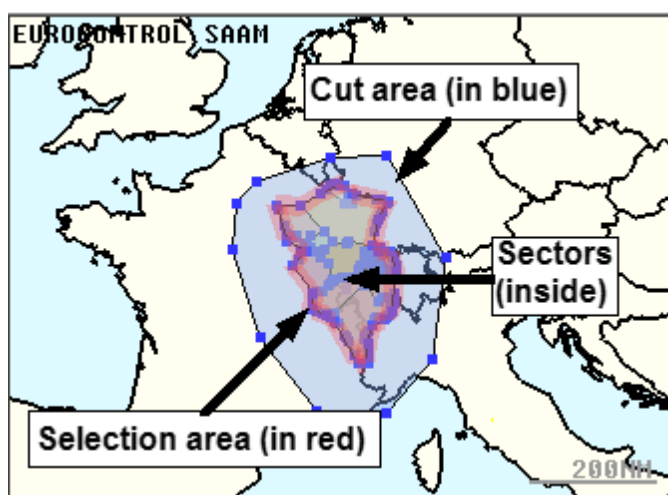
RAMS Scenario name

☐ Copy RAMS default files

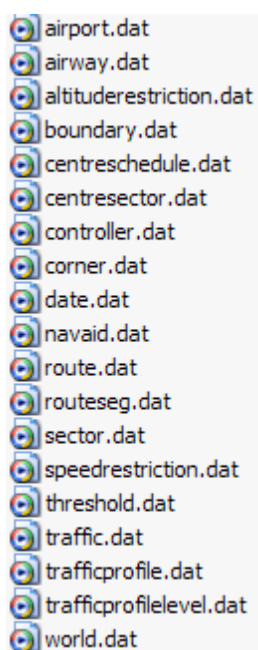
**RAMS Airspace Dialog Box**

## Parameters

1. **Input Traffic Trajectory:** *frame*– Browse an *so6* traffic file.
2. **Input Flight Level Constraint** *frame*– Browse a *flc2* FL constraint file (LOA).
3. **Input Traffic Selection Area:** \*.are and \*.sls or \*.gar and \*.gsl files containing ONE area. This area will be used to select all flights trajectories flying through. Generally this area is just around and very close to the external boundaries of all sectors to be simulated (a kind of external envelop).
4. **Input Traffic Cut Area:** \*.are and \*.sls or \*.gar and \*.gsl files containing ONE are. This area is used to cut the trajectories of selected traffic so that selected flights will start and end within this cut area. Generally the cut area is bigger than the selection area as it is roughly design around the Selection area with a convenient buffer in distance or in time.



5. **Input Airspace:** \*.are and \*.sls file containing the sectors that need to be simulated. Name of sectors should be as realistic as possible, preferably with 4 first letters aiming to designate ACCs.
6. **Sector configuration:** an optional cfg file can be browsed to provide RAMS sectors configuration information.
7. **Output RAMS directory** *frame* – Browse a name for the directory that will contained all generated RAMS files.
8. **Output RAMS scenario name** *frame* - type a name for the simulation to be exported in RAMS.
9. **Copy RAMS default files** checkbox – When checked (generally the first time a conversion is launched) all default RAMS files will be copied into the output RAMS directory (141 files), and some of these files will be changed according to the SAAM data files provided by this dialog box (19 files). In order to avoid re-setting ALL of these files during another conversion run (some files might have be changed by RAMS users preparing their simulations), user can un-check the box and only the files generated by the conversion will be created, other files will remain unchanged.



List of 19 RAMS generated files changed by SAAM data

## 2.9.4 Selected Airspace to Excel

**Accessed by (ONLY VISIBLE IF AN AIRSPACE IS SELECTED)**

**Menu Bar / Export Menu / Selected Airspace to Excel (ALT+X)**

### Purpose / Description

Converts already selected SAAM airspace, Airblocks or sectors, into Excel.

### Input

Selected airspace. Airspace can be selected from the [SAAM Airspace Editor](#), for example using:

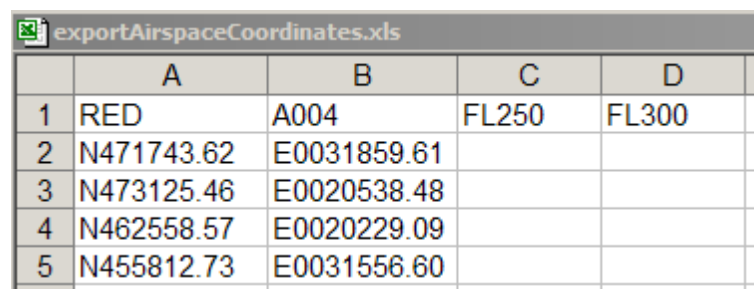
- shift + click on an airspace shape, or,
- Search button present on the Airspace Dialog Editor, or,
- when Airspace are pasted, or ...

### Output

Selected airspace are exported in a file called "exportAirspaceCoordinates.xls" located in the current working directory. Then, Excel is called to display this file.

Excel displays for each sector:

- a header made of one line (sector name, airblock name, airblock lower DFL, airblock higher DFL)
- a body made of several lines containing vertex coordinates (latitude, longitude)
- 



	A	B	C	D
1	RED	A004	FL250	FL300
2	N471743.62	E0031859.61		
3	N473125.46	E0020538.48		
4	N462558.57	E0020229.09		
5	N455812.73	E0031556.60		

### General Procedure

1. From the airspace editor select the airspace you want to export (See Input section above on how to select airspace)
2. From SAAM Menu, Export menu, click the sub-menu line "Selected Airspace to Excel (ALT+X)". Alternatively, one can use the shortcut: ALT+X, instead of using the Menu.

## 2.9.5 Network Points

### Accessed by

[Menu Bar / Export Menu / Network Points](#)

### Purpose / Description

Extract a list of points and their coordinates within a defined boundary.

### Input

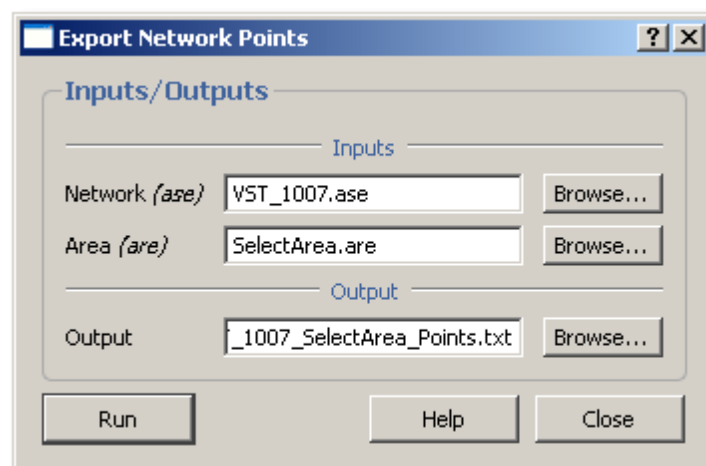
- Network (\*.ase)
- Airspace file (\*.are)

### Output

A text file containing a list of points and their coordinates. By default the file will be created in the scenario directory and its name is the concatenation of the two input file names.

### General Procedure

1. Specify the input network and the area.
2. Select the name and the location of the output file (optional).
3. Click on Run.



The export points dialog

### Parameters

1. **Input Network:** *frame* – Browse an ase network file.
2. **Input Area:** *frame* - Browse an are airspace file.
3. **Output:** *frame* - Select the name of the output text file.



## 2.10 Import Menu

### Accessed by

[Menu Bar / Import](#)

### Purpose / Description

This is a tool for converting one file format into another file format.

### SAAM to Unix Data Flow

Not yet Implemented

### RAMS Traffic

Not yet Implemented

### Access to DDR (internal)

You can use this utility to download data files from SAAM Demand Data Repository (DDR) using intranet (<http://prisme-oas.hq.corp.eurocontrol.int/ddr/>). Following data files can be downloaded:

- historical 4D Trajectory SAAM /NEVAC traffic files (built on CFMU data) and
- forecasted 4D Trajectory traffic files (built on SAAM Environment dataset files and STATFOR/FIPS dataset files).

To get access to DDR you have to register. Read more on the [DDR website](#)

### Access to DDR (external)

To down-load data files using SAAM Demand Data Repository (DDR) for external users.

### Data from Intranet (internal access)

Not more available

### Data from Extranet (external access)

Not more available.

### Get Basic Airspace Data

Will copy [Helicopter list](#), FIR data, ECAC Outer Area or Route Charge data into your working directory.

The FIR data are based on the data from Skyview dated Airac Cycle 9 April 2009.

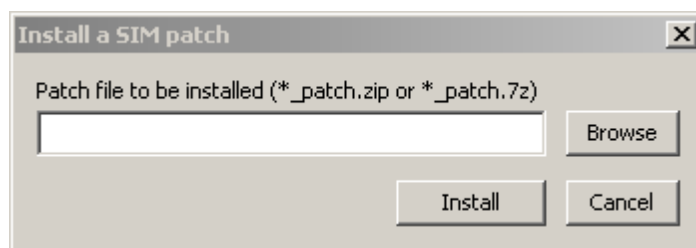
Route Charge data contains four files; two Airspace files (\*.are and \*.sls), Aircraft Weights file (\*.mwc) and CRCO Unit Rate file (\*.ur) and members.txt file. The CRCO Unit Rate file lists the unit rate for all member states in euro-cents.

## Install SIM patch

A patch is a file containing an extension or a modification of functionality.

This command will call a simple dialog box allowing to browse a patch file (which is a compressed file that must be called \*\_patch.zip or \*\_patch.7z). When browsed, pressing the "Install" button, will proceed to the installation of the new or updated functionality. It must be necessary to refresh SIM or re-launch SIM once patch is installed.

Caution: avoid to use a functionality while you are patching it, or a component of it. If a file to be patched is not accessible because you are using it, you will get an error message.



Dialog box to install a patch

### 2.10.1 RAMS

#### Accessed by

[Menu Bar / Import Menu / RAMS](#)

#### Purpose / Description

Converts RAMS data (flights and airspace) into SAAM data files for visualisation, edition or animation.

#### Input

- RAMS directory containing all RAMS data files, after a RAMS simulation was launched. In particular this conversion will read the RAMS file called "flightevent.out.1" and specific sectors RAMS data files (boundaries, corners ...).
- Date (for the traffic)

#### Output

- 4D trajectories (\*.so6)
- Sectors (\*.are and \*.sls, \*.gar and \*.gsl)

#### General Procedure

1. Select the location of the RAMS directory
2. Specify the name and the date output traffic file or uncheck the traffic output.

3. Specify the names of the airspace files or uncheck the airspace output.
4. Click Run.



**RAMS import Dialog Box**

## Parameters

1. **Input RAMS output directory:** *frame*– Browse a directory where RAMS has stored the result of its simulation.
2. **Output SAAM trajectories:** *frame*– Browse an so6 file
3. **Date of Traffic:** *calendar*– Pick a date that will be used for the SAAM traffic file. All flights generated in so6 format will start flying at this date. The limit of midnight is checked, and next date day might be stored where necessary in generated so6 file.
4. **Output Airspace Area:** \*.are and \*.sls or \*.gar and \*.gsl files that will contain sectors information build from RAMS data files.

### 2.10.2 Escape to SAAM

#### Accessed by

**Menu Bar / Import /Data from Escape Real Time Simulator**

#### Purpose / Description

The input files come from Bretigny's Real Time Simulator, Escape, and they are converted into SAAM files, like Traffic- and Airspace files.

## Input

- Traffic data, either in binary format or in text format.

Traffic data in binary format can be either Pre-simulation or Post-simulation. Both files contains a snap-shot of all flights every X seconds, with info for each flight lat/long, flight level, time etc... The point names in these files are fake. They are created 5 letters names. The binary file Pre-simulation file normally comes from RAMS and is called "navig\_RAMs". The binary file Post-simulation is normally called "navig\_3nav".

The text file contains info for all flights, flight level and time over route points. The point names in this files are real. There is also information about airports, type of aircraft and date that is needed for the data conversion. The text file is called Centr\_01....

- Environment data. A file that contains info about airblocks, sectors, TMAs and military areas active or not. This file is normally called "static.src"

## Output

- SAAM traffic files (\*.so6)
- Sector, Airblock, TMA files in Gasel format. Also Military areas, active or not, will be produced ( \*.gsl, \*.gar and \*.aco)

## General Procedure

1. Select Traffic Import, either Binary data or Text data. If Binary Data choose either Pre-simulation or Post-simulation or Environment Import
2. Browse the input files
3. Click on RUN

**Converting ESCAPE files into SAAM formats**

☐ Traffic Import - Binary Data (with Fake NAVAID names)

Data origin: ☒ Pre-simulation ☐ Post-simulation

Input 1 (no ext.):

Input 2 (select ext.):

Output (.so6):

☒ Traffic Import - Text Data (with Real NAVAID names)

Input (select ext.):

Output (.so6):

☒ Environment Import (Civil/Military Airspace)

Environment input:

Gasel output prefix:

**Data from Escape Real Time Simulator Dialog Box**

## Parameters

1. **Traffic Import - Binary Data (with Fake NAVAIDS names) frame** – Tick to activate
2. **Data origin radio buttons** - **Pre-simulation** or **Post-simulation**. Select one
3. **Input 1 (no ext.) field** – Binary file normally called "navig\_RAMs" or "navig\_3nav"
4. **Input 2 (select ext.) field** – Text file normally called "Centr\_01..."
5. **Output (.so6) field** – Will be named either "binary\_presimu.so6" or "binary\_postsimu.so6"
6. **Traffic Import - Text Data (with Real NAVAID names) frame** – Tick to activate
7. **Input (select ext.) field** – Same file as above for **Input 2**. Text file normally called "Centr\_01..."
8. **Output (.so6) field** – Traffic file named "text\_presimu.text"
9. **Environment Import (Civil/Military Airspace) frame** – Tick to activate

10. **Environment input field** - This file is normally called "static.src"
11. **Gasel output prefix field** - Gasel files with extension \*.gar, \*.gsl and \*.aco
12. **RUN** - Click to start the process.

## 2.11 View Menu

There is a possibility to change the look of the SAAM window. Under 'View' there are different options.

### Accessed by

[Menu Bar / View](#)

### Tool bar

Displays or hides the Tool bar

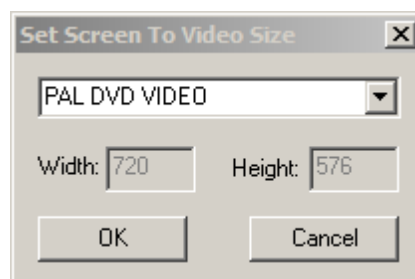
### Status bar

Displays or hides the Status bar

## Set Screen Size to Video Size

Opens a dialog box with different options to set windows dimensions that is suitable for a video or an image print. One option, called "CUSTOM" allows user to type its own size (in fields Width and Height). Other options are predefined window sizes and fields Width and Height, not editable, shows the dimension for the selected option. Available options are:

- PAL DVD VIDEO (720 x 576)
- NTSC DVD VIDEO (720 x 480)
- STANDARD SAAM MOVIE (912 x 604). Genrally used to create movies, this size is a good compromise between size and quality and fits well in a PowerPoint presentation (generally no need to resize).
- HD (1920 x 1080)
- HD\*3/4 (1440 x 810)
- HD\*2/3 (1280 x 720)
- HD/2 (960 x 540)
- CUSTOM. User choose Width and Height.



Dialog offering custom or pre-defined windows size

## **Set Top view**

Will change the camera position to top view

## **Set Side view**

Will change the camera position to side view

## **Set Level View Ref to ground**

Will set the Ground Level Reference Point back to ground

## **2.12 Help Menu**

### **Accessed by**

[Menu Bar / Help](#)

### **About SAAM**

Opens a window showing SAAM version and patch number

### **SAAM what's new ?**

All new features and changes to this SAAM version compared to the previous one are described here.

### **SAAM Reference Manual (HTML and pdf)**

Opens the on-line SAAM user reference manual in either HTML format or PDF format

### **SAAM Training Courses**

The PowerPoint presentations used in SAAM training course are available here

- Module 1 – Basic
- Module 2 – Scenario Processing
- Module 3 – Airspace Design

### **Small Video Training**

Contains the following tutorial video clips, in \*.avi format, to be played with 'Windows Media Player':

- How to Open a TDV file
- Mouse Zoom in
- Mouse zoom out
- Mouse Panning
- Mouse twist

- Mouse rotate
- Mouse view height
- How to make a query
- How to print an image and insert it in a document
- How to read Route in text format

## **SAAM Command line info**

SAAM can be launched by a shell or a DOS command prompt. Information about this can be found in [here](#).

## **Legend Set**

A set of labels, that can be useful when preparing a document using SAAM maps



# Chapter

---



3










## CHAPTER 3: SAAM Tool Bar

### 3.1 Tool Bar

#### Introduction

The Toolbar consists of several buttons. Some of these have icons which are well known to MS Windows users, whereas others refer to a specific SAAM function.

Buttons which duplicate items found in the Menu bar will not be explained here:

Icon	Functionality
	New
	Open
	Save
	Refresh
	Cut
	Copy
	Paste
	Print
	About SAAM

#### Edit TDV file



Opens the TDV file (\*.tdv) you are currently working with and let you edit it.

#### Toggle lighting for surfaces



Changes the appearance of hidden surfaces between solid and transparent.

#### Toggle smooth shading



Changes shading appearance for 3D volumes.

#### Toggle show negative volume



Allowing to change whether to display or not negative airblocks of sectors.

## Blue/White sea



Toggles sea colour between blue ( standard colour) or white.

## Set Sea Colour



Changes sea colour.

## Set Stereo Mode



Switches to stereo mode (special 3D polarized screen with associated viewing glasses are needed). If stereo mode is enabled and then if the TDV is saved the object "STEREO" will be present in the TDV.

## Set animation clock features



Regulates animation parameters.

## Swap 2D/3D



Toggles between 2D and 3D. When saving the 2D or 3D status of the current view will also be saved.

## Show presentation files



Interface to presentation. Displays the name and order of TDV files in the current presentation. Allows to select a particular TDV file.

“Auto Presentation” allows to automatically change of TDV from the list every X seconds.

If your TDV files contains animations, and “Auto Presentation” is set to zero seconds, there will be a change of TDV file when the animation is finished.

## Network info



Network info. Click on the **Network Info** button and the cursor will turn into an arrow with a question mark. If you then click on a segment you will get information about the segment points and the segment load.

## Aircraft Label



Shows aircraft level during animations.

## Editor



Sector/Sector/RAD Editor. Refer to SAAM Techniques chapter for more detailed explanations [Graphical Airspace Editor](#)



RAD/Rule editor. Several editors can be launched when pressing several times the button. When SAAM closes, they will be all closed (but each individual editor can be closed by clicking on its close button). [See Rule Editor](#)

## SAAM Scenario Dashboard



Not yet implemented

## SAAM Intuitive Maker (SIM)



Launch the [SIM](#) application (same as menu: Processing / SIM / Definition...)

## Airspace Configuration Editor



Airspace Configuration Editor. Refer to SAAM Techniques chapter for more detailed explanations [Config Editor](#)

## SAAM -> NEVAC communicator



# Chapter

---



4

## CHAPTER 4: SAAM Techniques

### 4.1 Dynamic Command of SAAM

#### Introduction

SAAM can be operated, in a limited way, from outside processes (even manually) in a very simple way. This possibility is given to advance users or to users having the capabilities to play with data, computers and programmes.

By default, SAAM is always listening (when opened !) for a specific text file called "dynamic\_command.txt" that must be located in "C:/SAAM\_USER\_PREF" directory (see the note below concerning the drive C:)

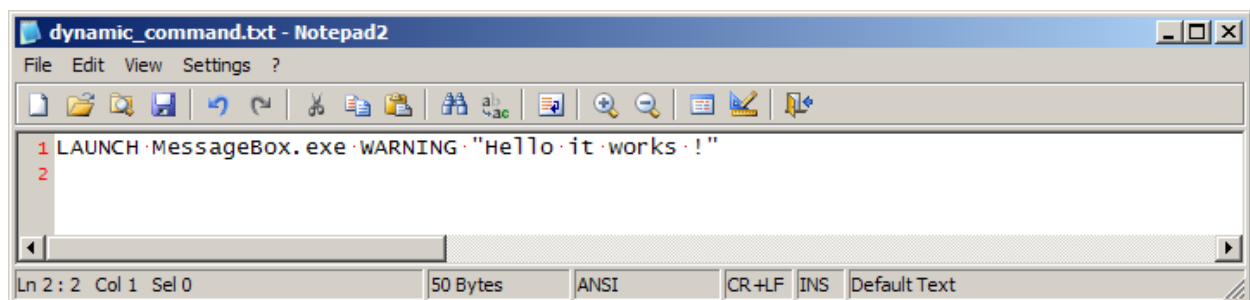
This text file "dynamic\_command.txt" can contains 2 commands which are REFRESH and LAUNCH (other commands are available but not documented because no yet validated). Theses 2 commands can have or not parameters (see below).

When the file "dynamic\_command.txt" containing the command is updated or written (in directory SAAM\_USER\_PREF) then SAAM will be reacting and will try to interpret the commands that it will find in it. Just before the command is executed, SAAM will delete the file "dynamic\_command.txt" automatically !

#### Possible commands:

- REFRESH (no parameter): SAAM will refresh (read again the content of the current TDV, possibly loosing what was un-saved).
- REFRESH file.tdv: SAAM will read and display the tdv file given in parameter (NO BLANK in tdv file name), tdv file must be present in the current working directory.
- REFRESH absolute\_path/file.tdv: SAAM will read and display the tdv file given in parameter (NO BLANK in path or in tdv file name), current working directory will be changed to the one where tdv file was found !
- LAUNCH file.exe parameters or LAUNCH file.ksh parameters: SAAM will try to launch file.exe or file.ksh with the accompanying parameters that are given. File.exe or file.ksh must be a command known by SAAM (so present in SAAM bin directory). Obviously the list of parameters that are given must be compliant with what file.exe or file.ksh is expecting.

#### Example:



**MessageBox.exe command requires 2 parameters**

When the text file "dynamic\_command.txt" is saved (in SAAM\_USER\_PREF directory)

SAAM (already opened) wakes up, launch the command, and delete the file !



**Result !**

## Note:

SAAM\_USER\_PREF directory is normally created automatically by SAAM if it does not exist, in C: drive. In some case, the computer administrator does not allow to create files or directories on C: drive, so another drive like D:/or C:/directory can be given (avoid remote drive of USB drive !!!).

See the value (and possibly change it) present for the key SAAM\_DEFAULT\_DIR in the text file "SAAM.ini" located in "C:\Documents and Settings\All Users\Application Data\EUROCONTROL". You can change this value when SAAM is OFF.

## 4.2 Using the Graphical Editors

### What is it used for?

The Editor allows the user to create or modify various types of airspace or networks.

### How do I start it?



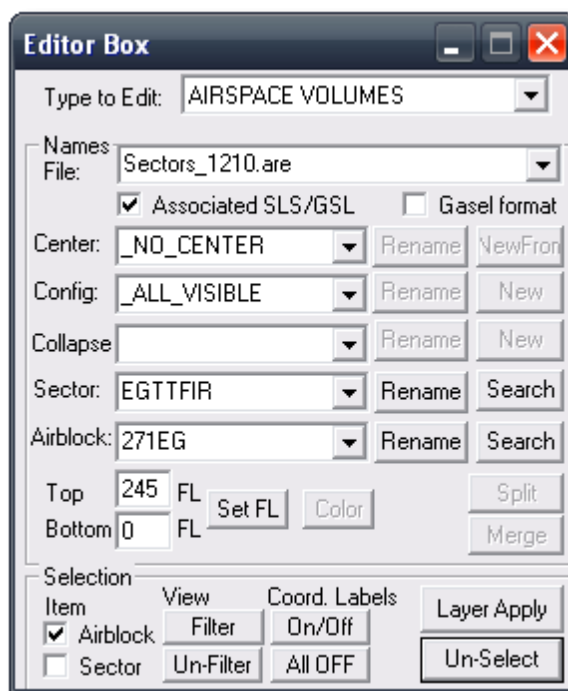
The editor is activated by clicking on 'Ed' button in the SAAM toolbar. This opens the Editor dialog box where the user may choose to edit **AIRSPACE VOLUMES** or **NETWORK**.

**Note:** The RAD/Rule editor can be opened from tool bar button



## 4.2.1 Editing Airspace Volumes

After selecting 'AIRSPACE VOLUMES' in the 'Type to Edit' box, new fields appear.



**Dialog box**

## Parameters

1. **Type to Edit drop-down box** - Select **Airspace Volumes**
2. **Names frame** – Information associated with the last selected airblock/sector.
  - a. **File drop-down box** – Provide for user information only the name and location of the Airspace file (\*.are or \*.gar) containing the last selected airblock/sector
  - b. **Associated SLS checkbox** – used to force creation of \*.sls file for old \*.are file having no associated \*.sls
  - c. **Config. drop-down list** – not yet implemented
  - d. **Center drop-down list** – not yet implemented
  - e. **Sector drop-down list** – Name of the last selected sector
  - f. **Airblock drop-down list** - Name of the last selected airblock
  - g. **Top/Bottom fields** – Will show the Top and Bottom level of the last selected airblock
3. **Selection frame** -
  - a. **Airblock checkbox** – Check to select one or several airblocks.
  - b. **Sector checkbox** - Check to select one or several sectors.
  - c. **Center. checkbox** – not yet implemented
  - d. **Layer Apply button** – This button is only available if modifications have been done on a layered airspace. It registers in the layer dashboard all the modifications done since the last time it was pressed. *Note:* Pressing the global save button does the same thing. See the [Layered Airspace Edition](#) section.
  - e. **Un-select button** – To unselect all selected airblocks/sectors.



**Important note:** When you are editing, don't forget to **save your Tdv** on a regular basis !!

## How to Edit Airspace

After activating the **Editor** you have four different edit modes to chose from using a combination of Group/Ungroup and 2D/3D view.

When the **Editor** is activated the cursor changes into a cross, indicating that the editor is operating in '**Ungroup**' mode. You switch to '**Group**' mode using '**ALT + g**'.

- **Ungroup** - Each airblock/sector can be edited separately. Only the selected object(s) are subject to an airspace edit change when **Ungroup** is used. By default, this mode is activated.
- **Group** - All volumes, selected or not, behind the mouse cursor will be subject to an edit action (move point, insert point).

Cursor



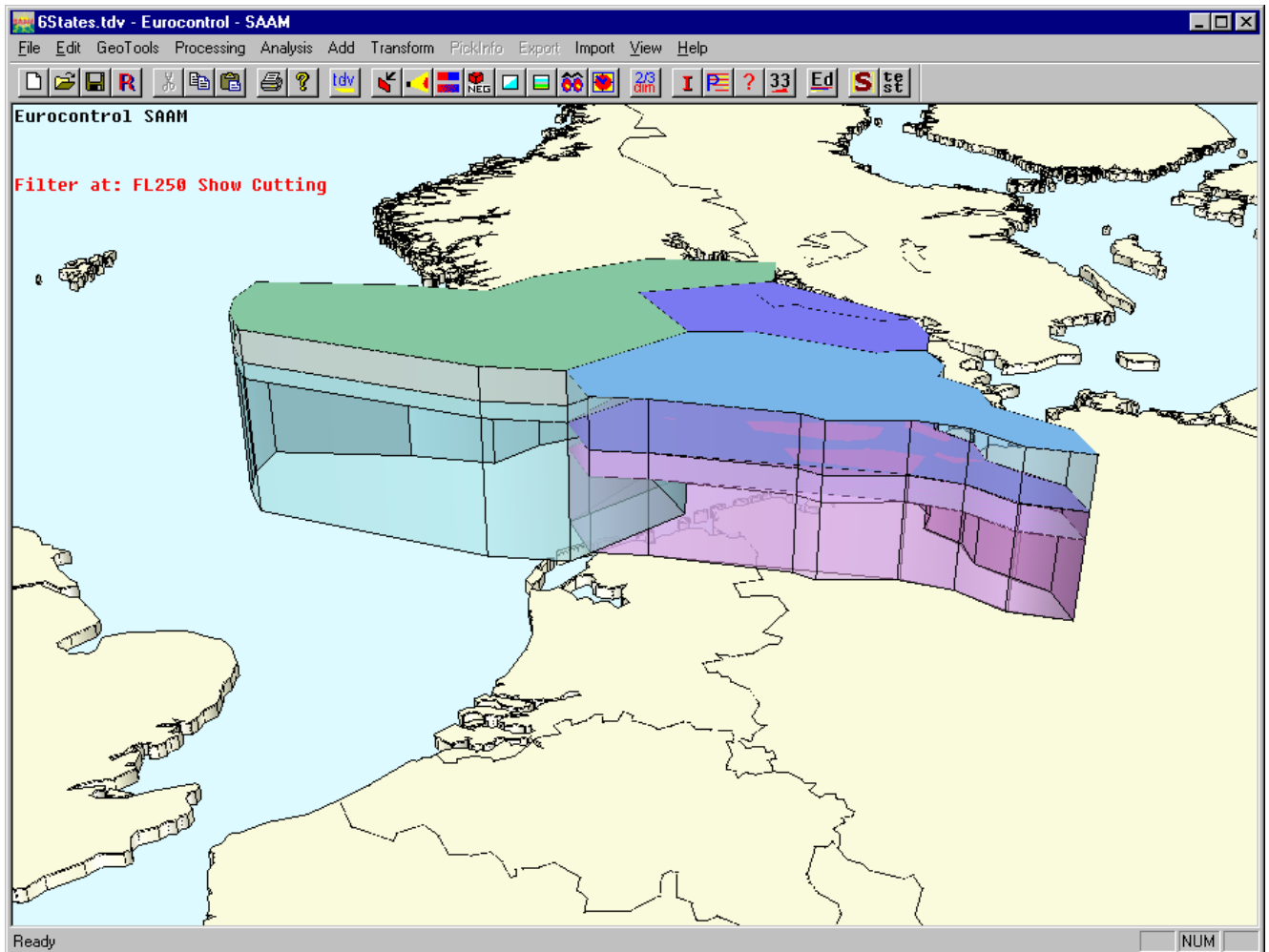
**Group**



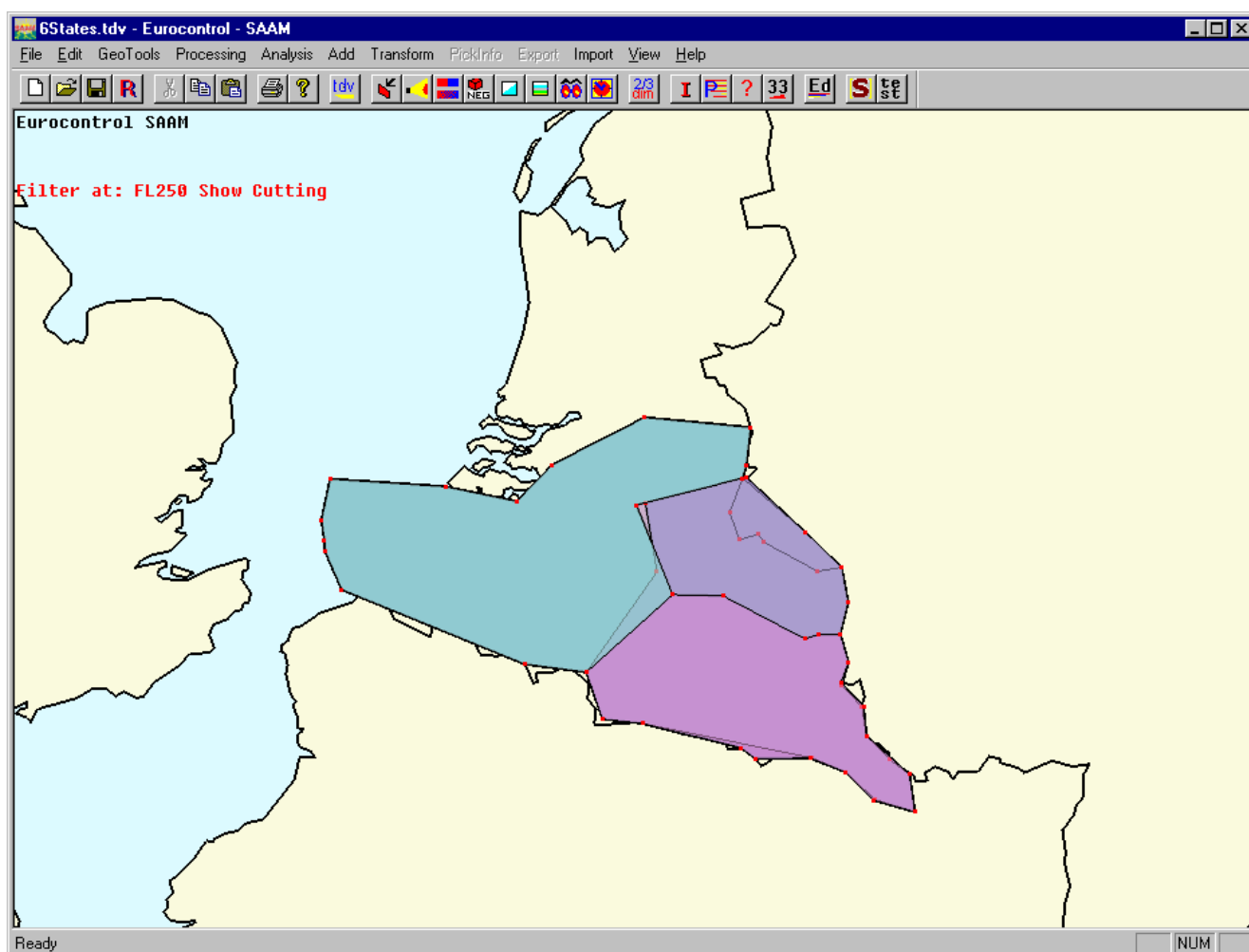
**Ungroup**

There are two different view modes, Edit and View mode, you switch between these by using keyboard '**Space Bar**'. Depending on the mode you can do different modifications.

- **Edit mode** - Displays the airspace in 2D and is used to view and edit areas (top & Bottom). The cursor is a cross with or without a G
- **View mode** - Displays the airspace in 3D and is used to view volumes and edit heights. The cursor is a an arrow with VIEW EDIT written below.



Edit mode



View mode

## Airblock and Sectors

In 'Airspace Editor' you work with airblocks and sectors.

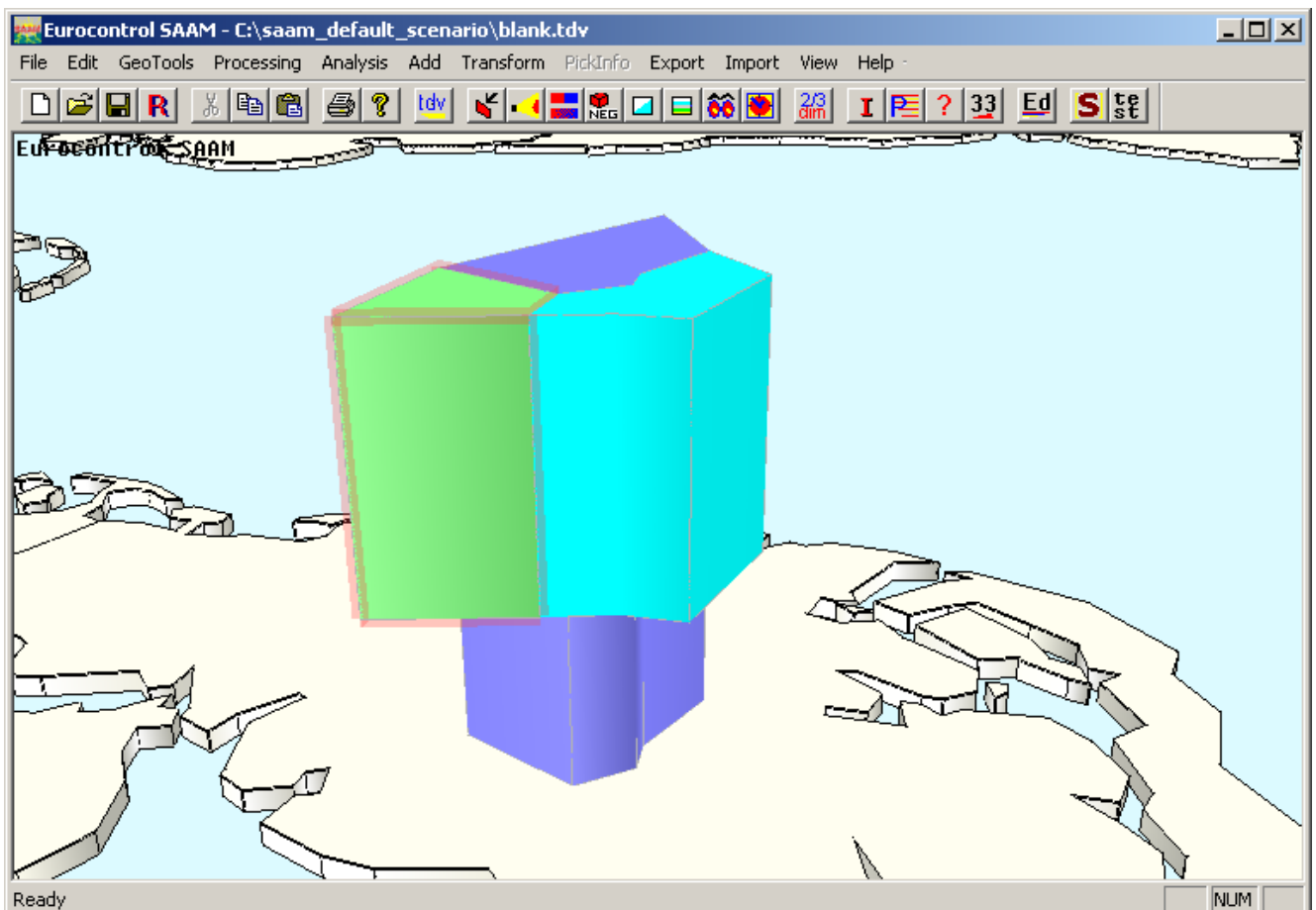
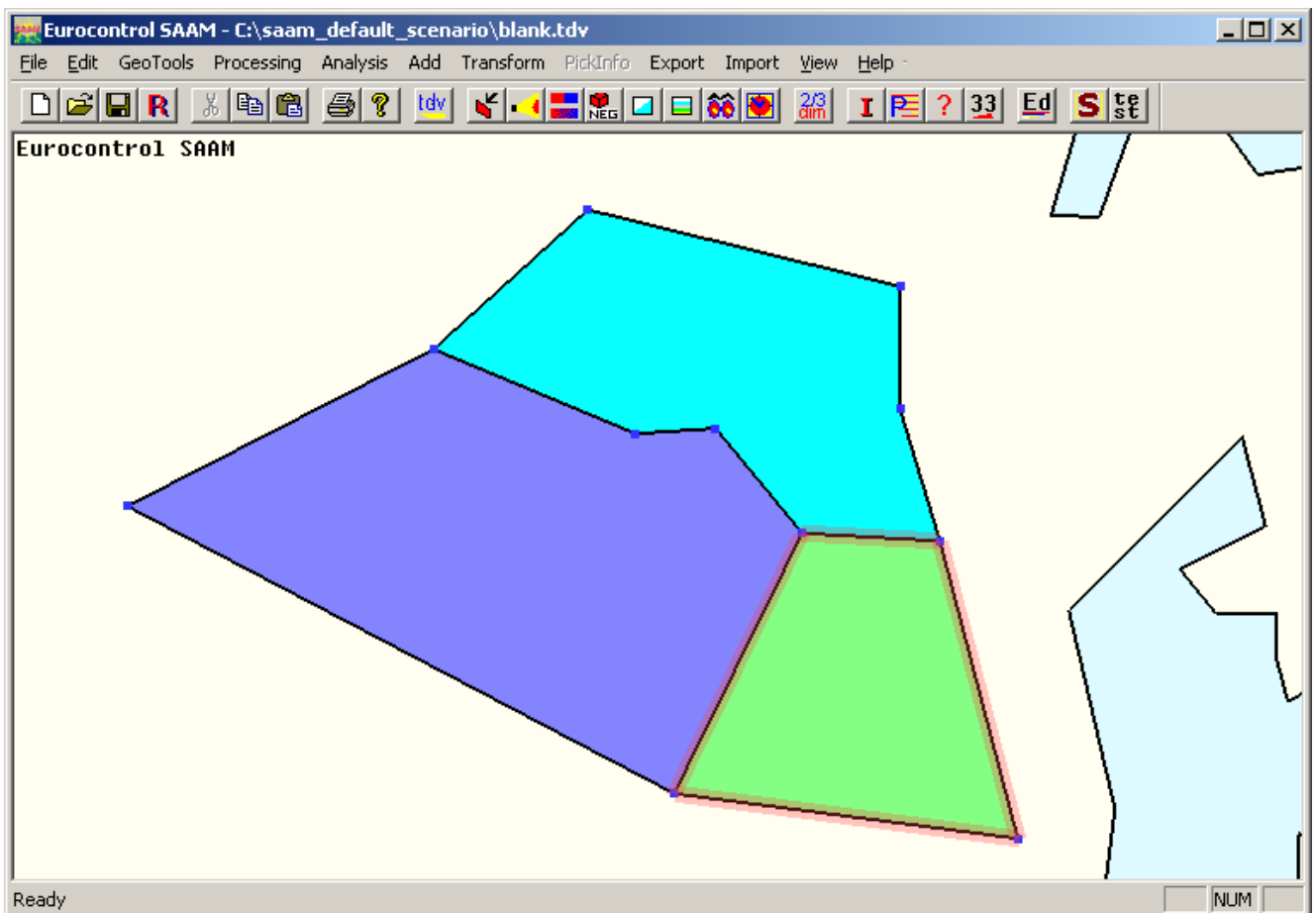
An airblock is a single two dimensional polygon, it is the basic element used when you build more elaborated airspace designs. It is the geographical coordinates that defines an airblock, not the altitudes. One airblock can belong to more than one sector, but with different bottom and top levels.

A sector can be defined as just one airblock (with a given bottom and top flight level ) or several airblocks grouped together.

## Select Airblock/Sector

Available in Group/Ungroup and Edit/View mode.

Position the cursor over the airblock/sector you want to select and press '**Shift + LMC** (Left Mouse Click)'. In order to select more than one airblock/sector just continue to hold down the Shift button and LMC on the next airblock/sector. The selected airblock(s)/sector(s) is displayed by a thick red transparent outer line.



There are two check-boxes in the Airspace Editor dialog box field '**Selection**' called '**Airblock**' and '**Sector**', these are used depending on whether you want to select an airblock or a sector. If '**Sector**' is checked all airblocks that belong to the chosen sectors will be selected

The '**Un-select**' button in the dialog box field '**Selection**' will un-select all your selected airblocks/sectors. If you don't want to un-select all of them, you just press **shift + LMC** on the selected airblock/sector and it will be un-selected.

## Select and View single Airblock/Sector

Available in Group/Ungroup and Edit/View mode.

Sometimes when you have a very complex map of airblocks/sectors it can be very difficult to view or to select the airblock/sector you want to.

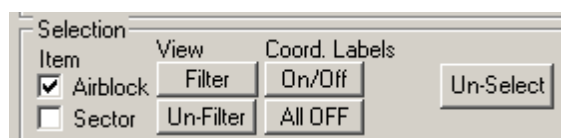
Use '**a + LMC**' if it is an airblock or '**s + LMC**' if it is a sector you want to select. The selected airblock/sector will be the only area displayed and all others will be hidden. This will make it easier to modify or view the selected airblock/sector. The background colour will change to light blue to indicate that there are hidden airblocks/sectors. Press '**n + LMC**' to go back to normal view.

There is a difference if you want to view a single airblock/sector in 2D or in 3D view. In 2D view you will still be able to see the outline and the points of all the other airblocks/sectors, but in 3D view they will be completely hidden.

It is only possible to edit airblocks/sectors that are displayed.

## View several selected Airblocks/Sectors

A graphical filtering can be activated based on selected airspace. Activation is made by clicking on button "Filter" located in Selection area. Only the selection will be seen, other airspace will be hidden, not deleted. Un-filtering can be activated by clicking on button "Un-filter" located in the selection area.



## Hide Airblock/Sector

Available in Group/Ungroup and Edit/View mode.

To help you viewing or selecting an airblock/sector you can hide, not display, the surrounding airblocks/sectors, by using '**Shift + a + LMC**' for airblocks or '**Shift + s + LMC**' for sectors. This action can be done repeatedly. Press '**n + LMC**' to go back to normal view.

## Move a Point

Available in Group/Ungroup and Edit mode.

Position the cursor over the point you want to move, **LMC and drag**.

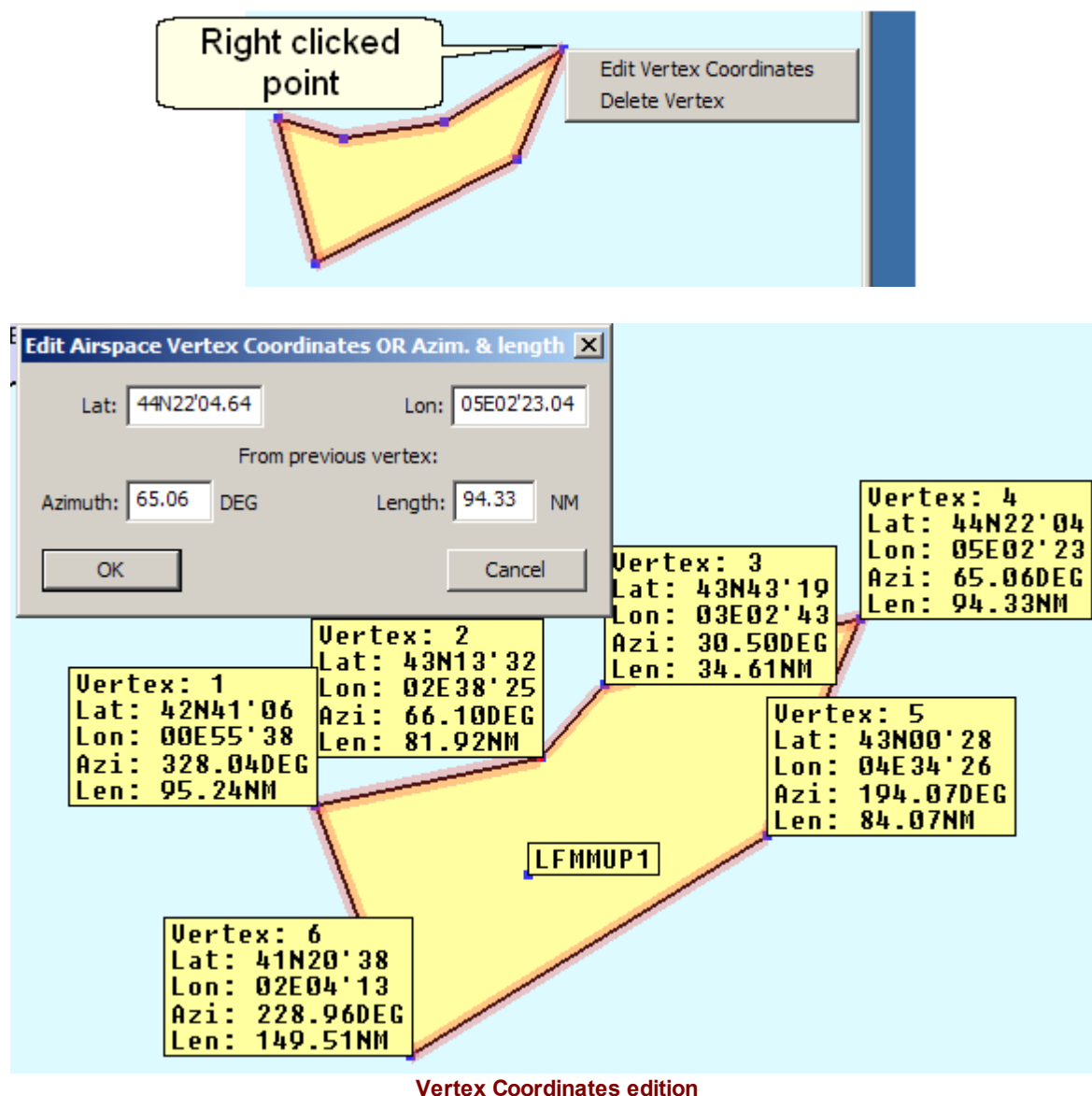
In '**Group**' mode, all airblock(s)/sector(s) to which point belongs behind the mouse cursor will automatically be selected. This is not the case with '**Ungroup**' mode, there you first have

to select the airblock(s)/sector(s) for which you want to move the point located behind the mouse cursor..

This operation can be used together with **'Where-am-I'** that is located in the menu bar under **'Geo tool'**. Then you will see the coordinates corresponding to the location of the point you are manually moving. You can also toggle the point labels **'ATL + K'** on selected volumes, and the coordinates will be displayed and updated during the manual move on their respective labels.

The snap function will automatically set the point on the location of the nearest already existing point, if not too far away, when the mouse button is released.

To set the exact coordinates of a point (instead of manual move), right click on a vertex to get a contextual menu offering to **Edit Vertex Coordinates**. This can be done by entering directly the coordinates or by entering length and azimuth from previous vertex (to know the vertex order use ALT + K, each vertex label has a number, the first vertex has number 1). If the point which is right clicked belongs to more than one polygons, the coordinates can not be set via length and azimuth, because it is obviously generally incompatible.



**'Undo'**, in the menu bar under **'Edit'**, is available for this operation.

## Inserting a New Point

Adds new points to one or more airblock(s)/sector(s).

Available in Group/Ungroup and Edit mode.

Position the cursor on the airblock(s)/sector(s) line and press '**Alt + LMC**'. After you can move the point to a desired spot.

If you work in '**Ungroup**' mode you first have to select the airblock(s)/sector(s) you want to add a new point to.

'**Undo**', in the menu bar under '**Edit**', is available for this operation.

## Delete a Point

Available in Group/Ungroup and Edit mode.

Select the point you want to delete by position the cursor over it and LMC. The point should be display with a small red dot, indicating it is selected. To delete it press '**Backspace**' from the keyboard.

Alternatively, and possibly preferably, you can also delete a point from contextual menu triggered with a right click above the point you want to delete. Select then from this menu "**Delete**".

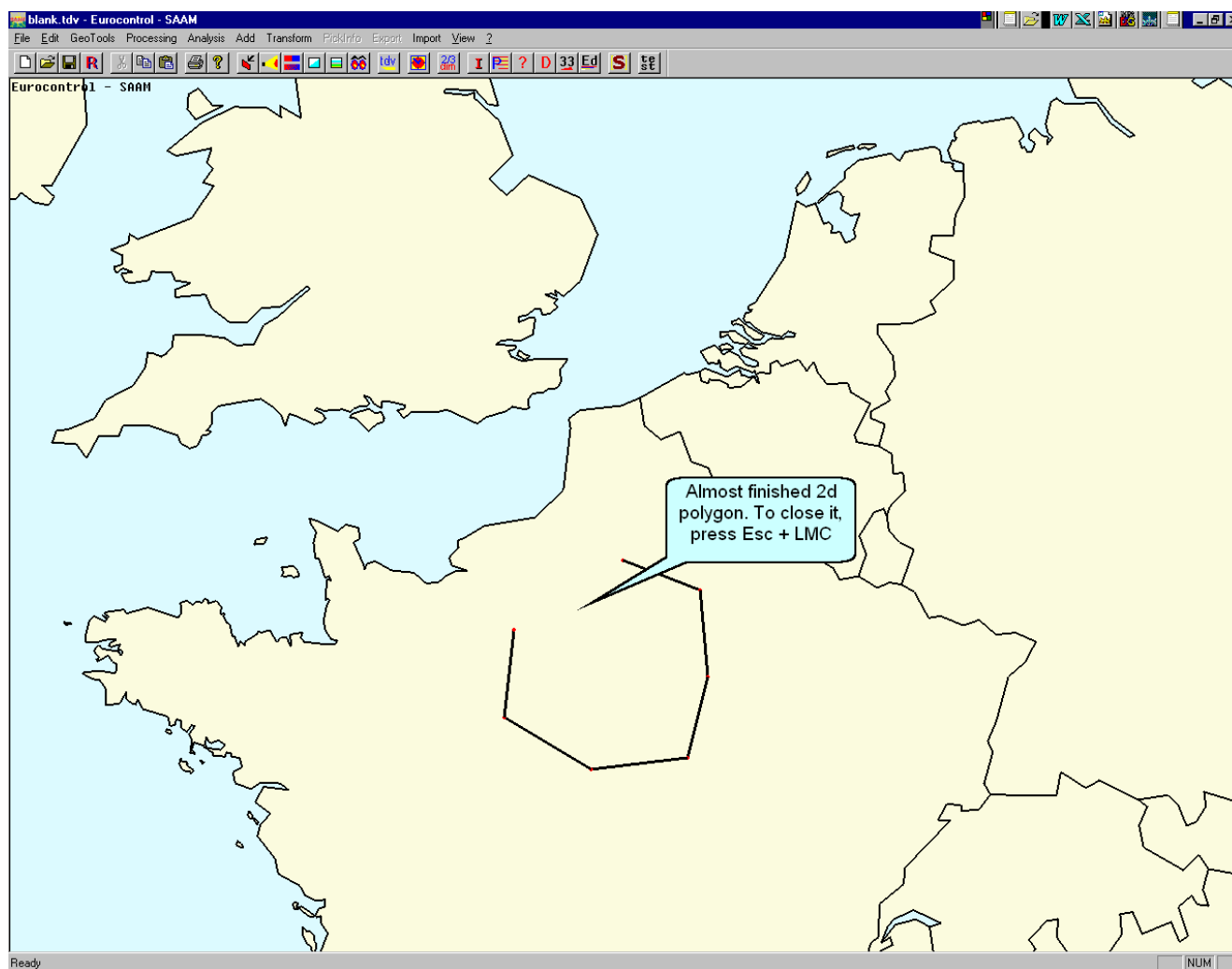
If you work in '**Ungroup**' mode you first have to select the airblock(s)/sector(s) you want to delete a point from.

'**Undo**', in the menu bar under '**Edit**', is available for this operation.

## Create Airblock/Sector

Available in Group/Ungroup and Edit mode.

To create an airblock/sector, first design a polygon by holding down the **CTRL** key and insert each point by left clicking the mouse. Insert a series of new points, preferably clockwise. Do not try to close the polygon manually by clicking on the first point, instead, simply press '**ESC + LMC**', the closure will be done automatically. The created polygon will by default be from FLo to FL100 and the names (airblock and sector) are chosen automatically by SAAM (starting with "A\_" for airblock and starting with "S\_" for sectors).



During the creation you can move the already created point

You will notice that the snap function will automatically locate the point (moved or created), when the mouse button is released, on the closest point, if it exist. This facility avoid making holes between polygons without changing the location of other points.

If you by mistake creates a new polygon, close it by pressing **ESC + LMC** and then delete it by pressing '**Backspace**' on the keyboard.

'**Undo**', in the menu bar under '**Edit**', is available for this operation.

## Delete Airblock/Sector

Available in Group/Ungroup and Edit/View mode.

Select the airblock(s)/sector(s) and press '**Backspace**' on the keyboard.

'**Undo**', in the menu bar under '**Edit**', is available for this operation.

## Change Division Flight Level

Allows the user to change the bottom or top DFL of a volume.

Available in Group/Ungroup View mode.

In 3D view, position the cursor on the top/bottom Airspace DFL line or corner you want to



change and press '**Alt + LMC**'. You can then drag the cursor, up or down, and the line will follow.

By default the step used to change DFL is DFL5 (so 500 feet per 500 feet), which is perfectly suiting airblocks and sectors. But if the SAAM airspace Editor is used to edit other type of Volumes (like a fake building or a height reference, etc ...) it is possible to change the DFL step by a much smaller value with the shortcut: Alt+o (with o from the main keyboard). Same shortcut is used to set again to default DFL step value (DFL5).

During the DFL change two small windows are displayed, a large one and a small one.

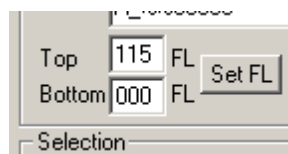
In the larger window you can see the level you are about to change and the name of the sector/s and airblock/s to which the line belongs. The position of the level in the window will tell you if it's a top level or a bottom level and if the level belongs to more than one sector.

The small window will follow the cursor and show the actual level of the line.

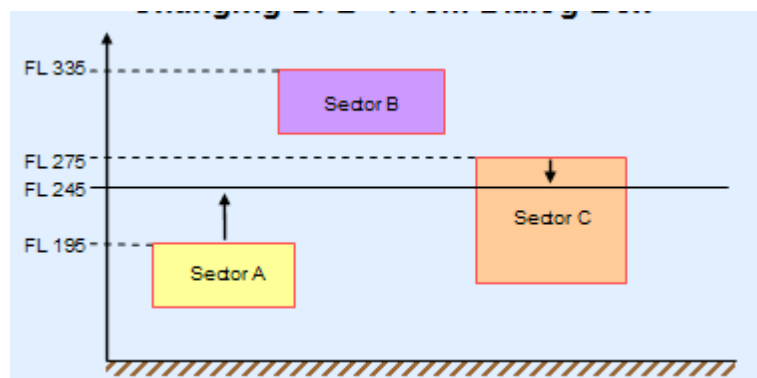
If you want to change the altitude of more than one airblock at the same time, you have to position the cursor over a common point or a common line.

If you click on a border side of a volume (instead of the top or bottom DFL line) then the entire volume will be lifted up or down.

Alternatively it is also possible, and sometimes more convenient, to change the DFL by typing the DFL value(s) in the editor dialog (in "Top" and/or "Bottom" window) and then click on button "Set FL".



From the editor dialog box you can change the level of several airblocks in same time, and independently the top, bottom or both levels. The airblocks doesn't have to be connected or even have the same DFL. Select the airblocks and type in DFL value(s) and then click on button "Set FL". A full numeric value check is made on the windows when values are input manually. Caution: if a given airblock taken from the set of selected airblocks can not cope with the bottom or top value that are requested, it will not be changed or it might become, at the extreme, a flat airblock. Undo is still possible.



**Sectors A, B & C are selected. Changing their Top DFL to FL245 will work for sectors A & C, but will not work for sector B !**

'**Undo**', in the menu bar under '**Edit**', is available for this operation.

## Display Labels

Available in Group/Ungroup and Edit/View mode.

When you press '**Alt + d**' (d like Designator) an airblock/sector label will be displayed. First time you press '**Alt + d**' there will just be one label per sector with the sector name. If you press '**Alt + d**' again you will get labels for all the airblocks with information about the sector name, airblock name, top and bottom levels. Pressing '**Alt + d**' once more will hide the labels. You will notice that labels will appear in the middle (vertically) of the volumes if they are transparent, else above them (case they are solid).

**No label => ALT + d => Sectors Label => ALT + d => Airblocks Labels => ALT + d => No Label**

## Change Colour

Available in Group/Ungroup and Edit/View mode.

Once airblock(s)/sector(s) is/are selected you can change the colour by pressing '**Alt + C**'. A colour palette will open and from there select the colour you want.

'**Undo**', in the menu bar under '**Edit**', is available for this operation.

## Change between Solid and Transparent

Available in Group/Ungroup and Edit/View mode.

To change airblock(s)/sector(s) from solid to transparent, or vice versa, first make a selection and then press '**Alt + t**'. You will notice that sectors or airblocks labels are displayed in the centre of the volume if this last is transparent, or above the top surface of the volume if this one is solid.

'**Undo**', in the menu bar under '**Edit**', is available for this operation.

## Change name of airblock/sector

After an airblock/sector is selected the name of it will appear in the Airspace Editor dialog box. There you can see both the sector name and the airblock name. To change one of the names, type the new name in the appropriate field and then press the button 'Rename' next to it. You can not change sector and airblock name in same time. You will notice that airblock names are chosen by the computer are generally not relevant for the operational side of your work (technically the name is important because an airblock can be shared between several sectors, especially the stacked ones), while sector name is quite important because it will appear in your analysis. Generally it is better to let the computer choosing airblock names.

Note: sector name must have maximum 19 characters and airblock names must have maximum 24 characters.

'**Undo**', in the menu bar under '**Edit**', is available for this operation.

## Group several airblocks or sectors into one sector

A sector can consist of several airblocks. To group airblocks (or even several sectors) into one sector you must give them the same sector name. This can be done one by one or you can do

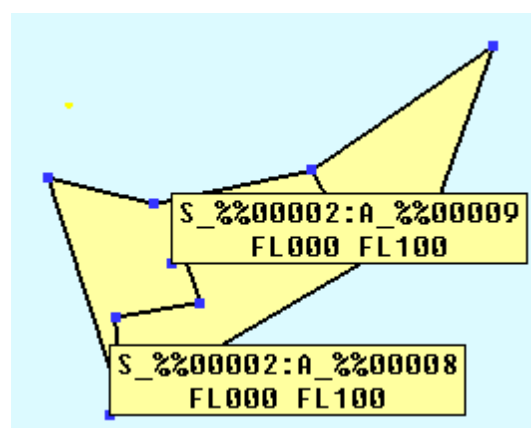
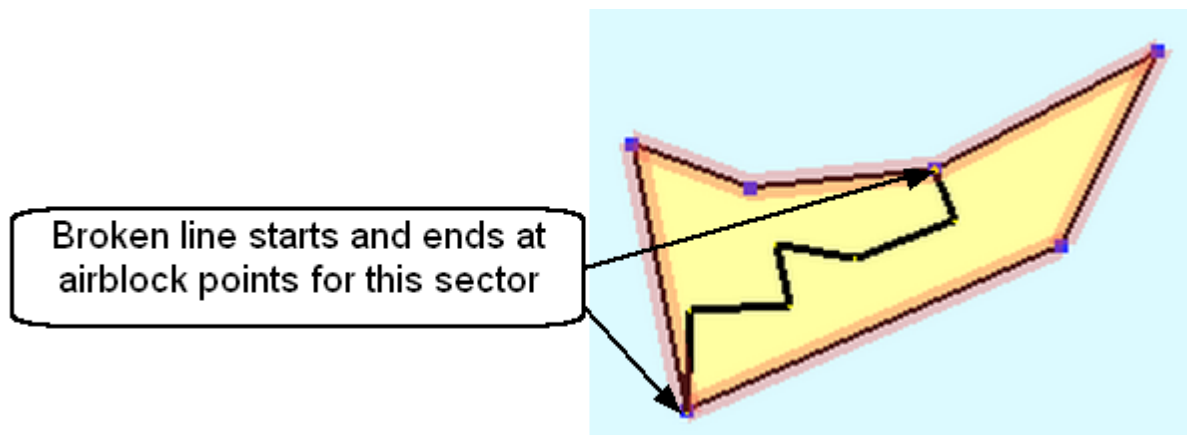
it all at once. To do them all at once, select all the airblocks and change the sector name in the Airspace Editor dialog box. The advantage to do them all at once, is that they will all get the same colour, the colour and the name of the last selected airblock.

'Undo', in the menu bar under 'Edit', is available for this operation.

## Split Airblocks

This functions gives you the ability to split an airblocks in 2 pieces according to a broken lines introduced by the user. Several stacked airblocks (not necessarily having the same shape, nor to be exactly touching on top of the other !) can be operated with split in one time.

The broken line is introduced in the same way as airblock creation, using CTRL + LBC. The first and last point of the broken line must be over points of the selected airblocks subject to the split.



**Result: we have now 2 airblocks for this sector**

You will notice that is the broken line runs outside the airblock to be split the result, still valid, could be surprising.

The button "Split" in the editor dialog will automatically available if the conditions stated above are filled.

The resulting airblocks are replacing the original ones and have the same features (color, transparency, name ...) !

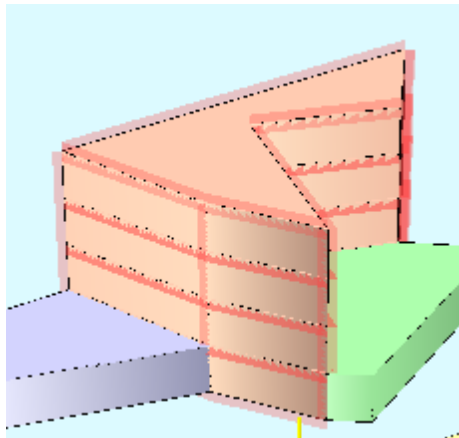
‘Undo’, in the menu bar under ‘Edit’, is available for this operation.

## Merge of airblocks

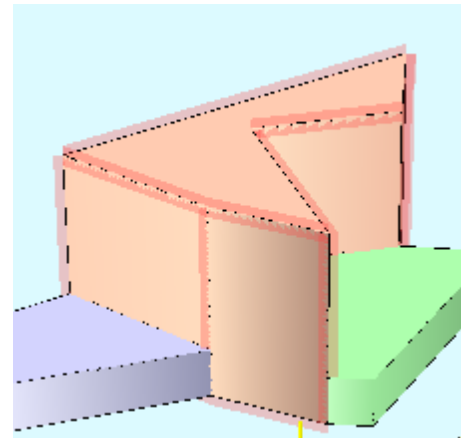
This functions gives you the possibility to merge, vertically or horizontally, connected airblocks (side by side or top to bottom surface). The result will be a new single airblock, replacing the selected ones, having the total volumes.

The button “Merge” in the editor dialog will be available once the conditions to merge are possible. For instance isolated airblocks will not be merged. A maximum of 400 airblocks can be merged. The function to detect merge possibility (and enabling or not the “Merge” button) is working with normal airblocks selection, point move, search, insert point.

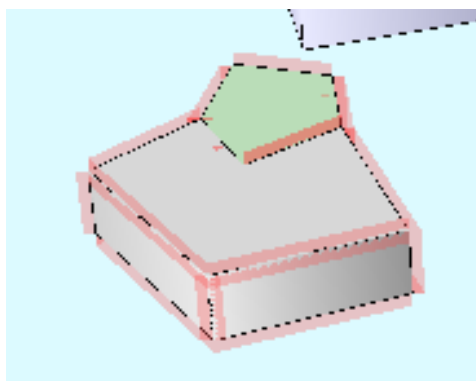
‘Undo’, in the menu bar under ‘Edit’, is available for this operation.



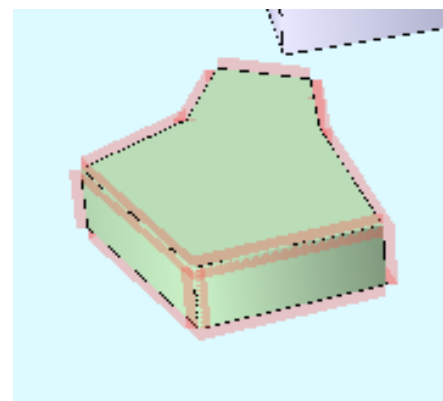
*Example of vertical merge before*



*Example of vertical merge after*



*Example of horizontal merge before*



*Example of horizontal merge after*

## Search for Airblock(s)/Sector(s)

In the Airspace Editor dialog box you have the possibility to search for one or several

airblocks/sectors. Type the name of the airblock or sector in the appropriate field and the press **'Search'**. The found airblock/sector will be selected and displayed in the centre of the viewport.

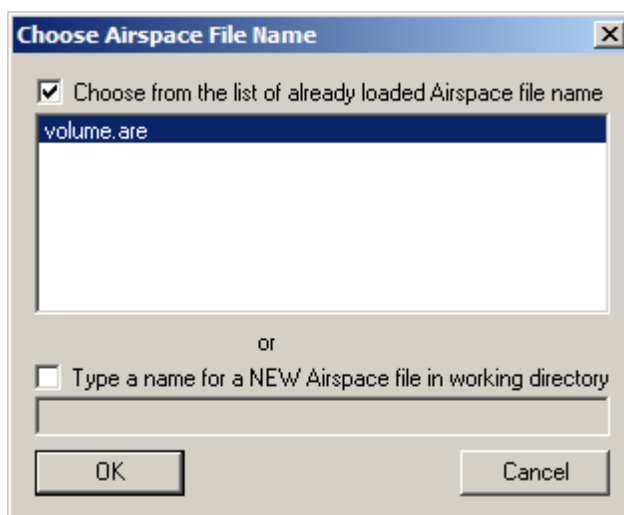
You can also use wildcards to expand your search. An asterisk (\*) will replace any number of characters at the end of a search string.

This is useful to select and copy/paste a given center to be worked out separately.

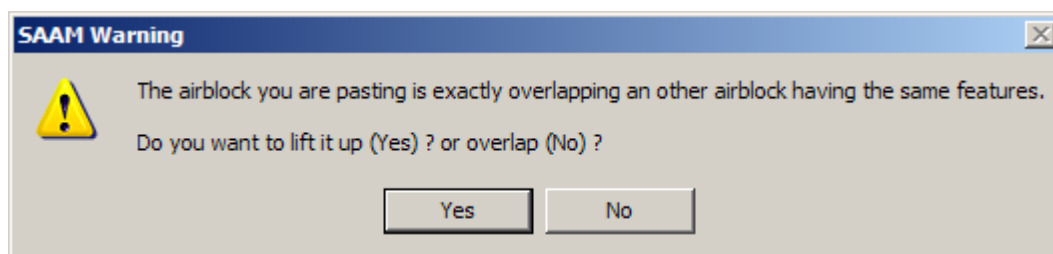
## Copy and Paste airblock/sector

You can copy and paste one or several airblock(s)/sector(s) at a time, either into an already loaded Airspace file (\*.are) or into a new one. Copying must always be done from the Airspace editor, while paste can be done directly outside the editor.

Select the airblock(s)/sector(s) and under **'Edit'** in the menu bar select **'Copy'** and then select **'Paste'**. A new dialog box will open, where you have the options to choose into which Airspace file (\*.are) you want to paste the airblock(s)/sector(s). This gives possibility to split your airspace in different data files.



If there is just one airblock you want to copy from one Airspace file (\*.are) and paste it into the same file, SAAM will detect an overlap and then will ask if the airblock have to be lifted up above the copy, or simply duplicated. This function is nice to quickly build stacked sectors.



**Note:** If there are more than one airblock or a sector pasted into the same Airspace file (\*.are), as they were copied from, they will be pasted overlapping.

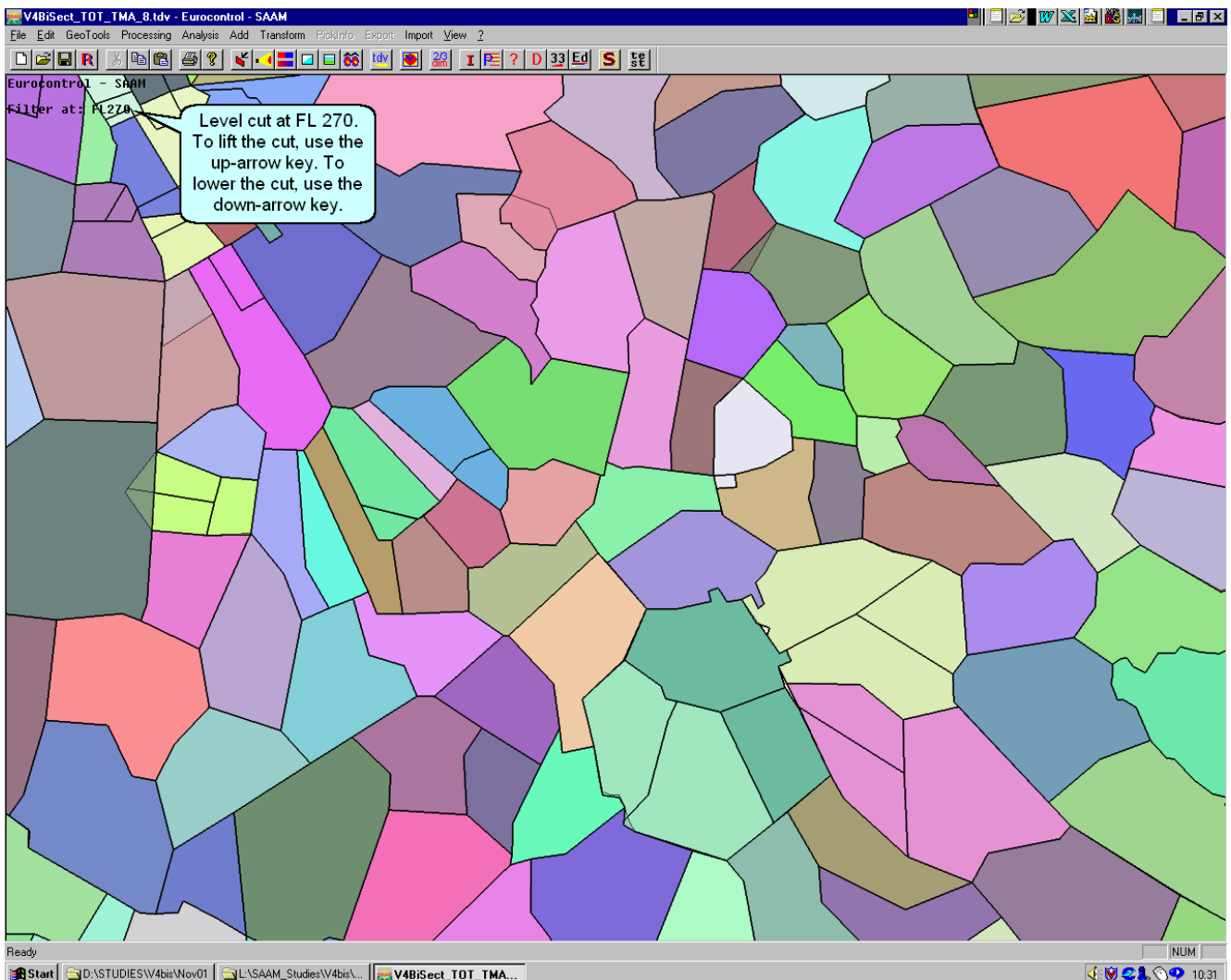
## Select FL Slice

Makes it easy to find holes or overlaps in the sectorisation. It can also be useful for regular airspace editing.

Available in Group/Ungroup and Edit mode.

Press '**Alt + f**'. The cut is displayed in the top left corner. To change it, use the arrow keys up and down on the keyboard. It will view a slice of the airspace every 1.000ft. (FL10 step).

It is only possible to edit airblocks/sectors that are displayed.



## Double Cut

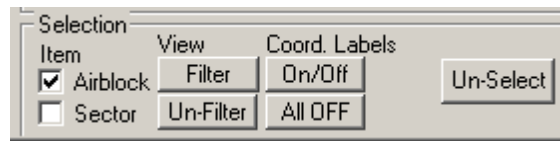
A **Double Cut** function is also implemented. To activate it use the right arrow key. Two level cuts will appear. The one that is in red can be edited via the up or down arrow keys ; the right arrow key acts as a toggle between the editable level cuts.

To go back to only one level cut, merge the two levels together then press once the right arrow key.

## Coordinate Labels for Vertices

The shortcut ALT+K will enable/disable display of coordinate labels for selected airblocks. It can also be activated by clicking on "On/Off" button for Coordinate labels. The button is

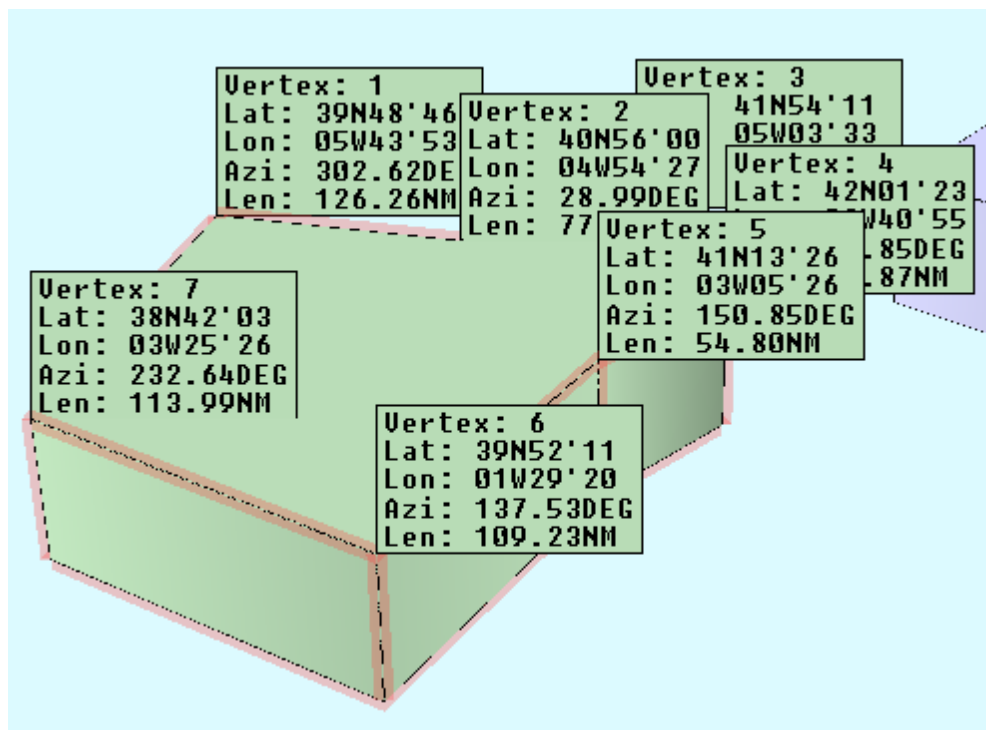
located in Selection area. There is another button "All OFF", that will delete all coordinate labels.



A point label contains the sequence order of the point for the polygon, the coordinates, and the length and azimuth from previous point. The background of the label has the same color has the airblock.

Labels are automatically updated when points are moved, inserted, color changed, airblock label moved ...

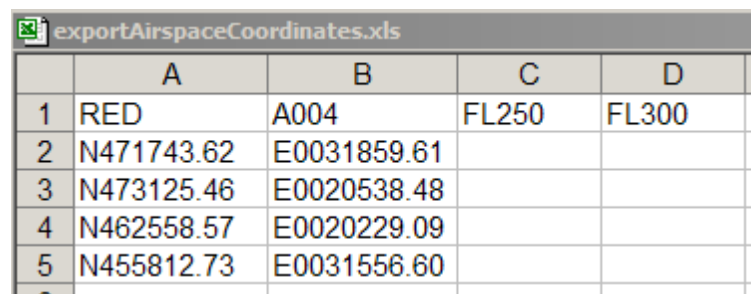
The location of the point labels depends on the location of the airblock label (normally somewhere in the center).



## Export Airspace Coordinates into Excel

Selected Airspace from Airspace Editor can be exported to a text file that will be automatically opened by Excel (the name of the text file is "exportAirspaceCoordinates.xls" and is located in your current working directory). The command to launch the export function is either the shortcut ALT+X or via the SAAM menu bar under "Export", "Selected Airspace to Excel" (see [Export Menu](#)). You will notice that this menu command is visible only if at least one airspace is selected (so subject to be exported).

If several airblocks are selected, whatever the selection order, they will be grouped per sector name within Excel sheet.



	A	B	C	D
1	RED	A004	FL250	FL300
2	N471743.62	E0031859.61		
3	N473125.46	E0020538.48		
4	N462558.57	E0020229.09		
5	N455812.73	E0031556.60		

Example of a 4 vertices sector called "RED" running from FL250 to FL 300 (Airblock is called "A004")

## Graphical Slice Cut

Available in Group/Ungroup and View mode.

‘**Alt + s**’ will give you the possibility to zoom in and look at the graphics from the inside. The words ‘Eurocontrol SAAM’ in the top left corner of the viewport disappears when ‘Slice cut’ is active. To deactivate it you just press ‘**Alt + s**’ once more.

It is only possible to edit airblocks/sectors that are displayed.

## Converting SAAM format to/from Gasel format

When an airspace item is selected, its features are displayed in the dialog box (name, DFL, data file ...) but also its current format in the check box "Gasel format". If this check box is off the format is SAAM (default) and next saving will update or create are/sls files. If it is on, then next saving will update or create an gar/gsl files.

This allow to convert one format into another one without deleting original format.

Note: ".are" file are expressed in minutes decimals with 4 digits, while ".gar" file are expressed in degrees decimals with 6 digits. The accuracy is not exactly the same. An Airspace Traffic Intersection might not give the same ".t5" file for SAAM format and for Gasel format !

## Lock 3D Airblock/Sector

Locks an airblock/sector, making all subsequent modifications impossible for that airblock/sector.

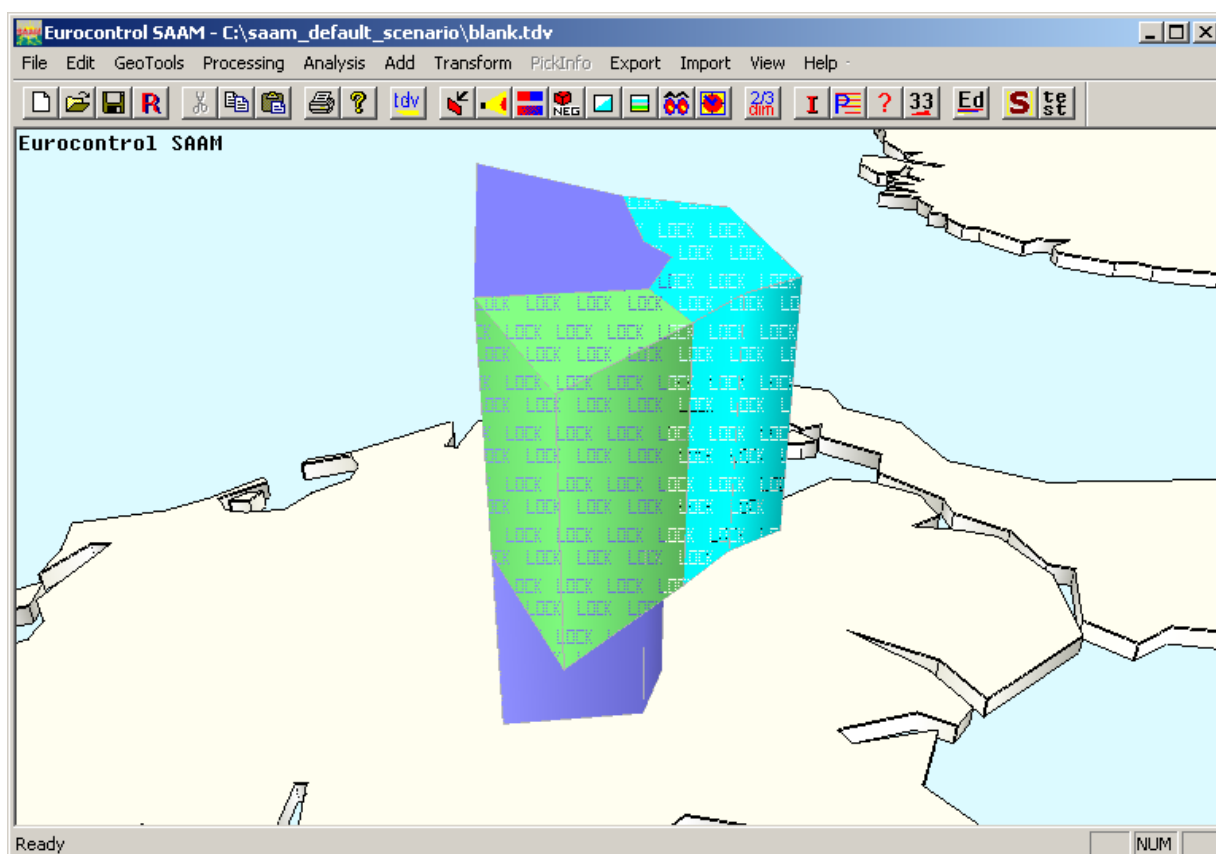
Available in Group/Ungroup and Edit/View mode.

First you have to select one or more airblock(s)/sector(s), then you press ‘**Alt + l**’ to lock them (**l** like in **lock**). To unlock you have to press ‘**Alt + l**’ again.

Another possibility to lock an airblock/sector is to write the airblock/sector name in uppercase i.e. LOCKED\_AIRBLOCK.

A locked airspace will be recognised by LOCK written in it.





## Layered Airspace Edition

It is possible to edit an airspace date inside a layer. In order to do that, you need to add a layered file (generally a 'oooo' file) in your TDV.

When you edit an airblock or a sector of your layered file, the button "Layer Apply" will appear in the bottom right corner of the editor. Clicking it will register all your modifications inside the layer dashboard.



The "Layer Apply" button in the airspace editor

### Limitations:

- The edition of layered airspaces is only compatible with ARE/SLS airspace files. GAR/GSL format are not yet supported.
- When you copy/paste an airblock or a sector, check that the id of the new sector doesn't already exist in the file. If it does, the created layer modifications in the dashboard will become red the next time the consistency checks will be performed. i.e. when you change the view or when you refresh the current view (from the layer dashboard: View -> Reload Current View).
- There are some known issues with airblock renaming. If you need to rename an airblock,

try to perform other actions before hitting the "Layer Apply" button, and reload the current view (from the layer dashboard: View -> Reload Current View) to check if your modifications are valid and have been taken into account.

**N.B.**

Once your volume is edited or even during edition, don't forget to **save** your Tdv file !!!

## 4.2.2 Editing Airspace: Commands & Shortcuts summary

### Edit Airspace commands and shortcuts

Below are the Airspace Volume Edit commands and shortcuts available (" V" means command is valid for the different modes)

Airspace Volume Edit operations	Group	Ungroup	Edit mode	View mode	
Select Airblock/Sector	V	V	V	V	Shift+LMC (Left Mouse Click)
Select vertex	V	V	V		LMC on the vertex and move
View Only clicked single Airblock/Sector	V	V	V	V	a+LMC or s+LMC
Hide Only clicked Airblock/Sector	V	V	V	V	Shift+a+LMC or Shift+s+LMC
Move Vertex	V	V	V		LMC on the vertex and move
Insert New Vertex	V	V	V		Alt+LMC on a segment
Delete Vertex	V	V	V	V	Backspace
Create Airblock/Sector	V	V	V		Ctrl+LMC, Esc+LMC to close
Delete Airblock/Sector	V	V	V	V	Backspace
Change Altitude (graphically)	V	V		V	Alt+LMC on a DFL border and move up/down
Display Labels	V	V	V	V	Alt+D
Change Colour	V	V	V	V	Alt+C
Change between Solid and Transparent	V	V	V	V	Alt+T
Select FL Slice	V	V	V	V	Alt+F or F9
Graphical slice cut	V	V		V	Alt+S
Lock Airblock/Sector	V	V	V	V	Alt+L
Export Airspace to Excel	V	V	V	V	Alt+X
Display vertex Coordinates	V	V	V	V	Alt+K
Swap DFL change step	V	V	V	V	Alt+0 (0 from main keyboard)
Translate airspace (no undo)	V	V	V	V	T and mouse click and move
Rotate airspace (no undo)	V	V	V	V	R and mouse click and move
Activate Group		V	V	V	Alt+G
Activate Ungroup	V		V	V	Alt+G
Activate Edit Mode	V	V		V	space bar
Activate View Mode	V	V	V		space bar

### 4.2.3 Editing Route Network

When selecting '**NETWORK**' in the '**Editor Box**' you can either adjust the parameters to display the network or edit it.

#### View / Edit mode

When you are in the Network editor there are two different view modes, Edit and View mode, you switch between these by using keyboard '**Space Bar**'.

- **View Mode** – Cursor is displayed as an arrow with VIEW EDIT written below. Camera position is un-locked: zoom and panning are possible, but only in top view. Network can not be changed
- **Edit mode** – cursor shape is displayed as an “up arrow”. Camera position is locked (no zoom, no panning). Mouse buttons and movements are used to change the network

#### Displaying the Network

There are different ways the segments can be displayed. Using colour codes makes it easier to get a good overview of the network.

**Alt + n** toggles between the different displays:

**Type** – The network colours shows which the type of segment it is. A **Normal** segment is grey, **Arrival** is pink and **Departure** is gold.

**Parity** – The network colours shows the parity of the segment. **Odd** is green and **Even** is blue.

**Mixed** – The network colours is a mix between **Type** and **Parity**. The **Normal** (grey) segments in **Type** have changed colour to show **Parity** (green and blue). **Arrival** and **Departure** are displayed as in **Type**.

**Default** – The network coloured with the default colour

**Load** – The network colours shows the **Load** of a segment depending on the colour settings and the thresholds.

**CDRs** – Only the CDRs are displayed.

#### Editing the Network

The network can be edited in all the different displays.

Be careful when editing in **Load** display. A newly created segment will have zero as segment load, and may not be displayed because of the thresholds.

The 'Editor Dialog box for Network' is a software window with a title bar 'Editor Box'. It contains several input fields and buttons. At the top, 'Type to Edit' is a dropdown menu set to 'NETWORK'. Below it is a 'File Name' dropdown menu showing 'mod\_3.ase'. The 'Item Name' field contains 'SUBES', with 'Change' and 'Search' buttons to its right. The 'Lat' field shows 'N472506.60' and the 'Long' field shows 'E0172552.20', with a 'Change' button to the right of the 'Long' field. A 'Locked' checkbox is checked. The 'CDR' dropdown menu is set to 'NORMAL', with a 'CDR setting' button to its right. The 'Parity' dropdown menu is set to 'ODD (green)' and the 'Type' dropdown menu is set to 'NORMAL (grey)'. Below these are three input fields: 'Size: Point' (0.50), 'Seg.' (1.00), and 'Arrow' (0.80). A 'Labels' section contains a 'Point Name' checkbox, a 'Load Value' checkbox, and a table for 'Nb seg: Converg. Diverg. Nb Decimals' with 'Max' and 'Min' values and 'load' buttons.

**Editor Dialog box for Network**

## Parameters

1. **Type to Edit drop-down list** - Select '**NETWORK**'
2. **File name drop-down list** - If you have a Tdv file opened that contains network file/s (\*.ase) it/they will be displayed here.

The 'Create a Network' dialog box has a title bar 'Create a Network'. It contains a label 'Network ASE File Name:' above a text input field containing 'test\_2'. At the bottom are 'OK' and 'Cancel' buttons.

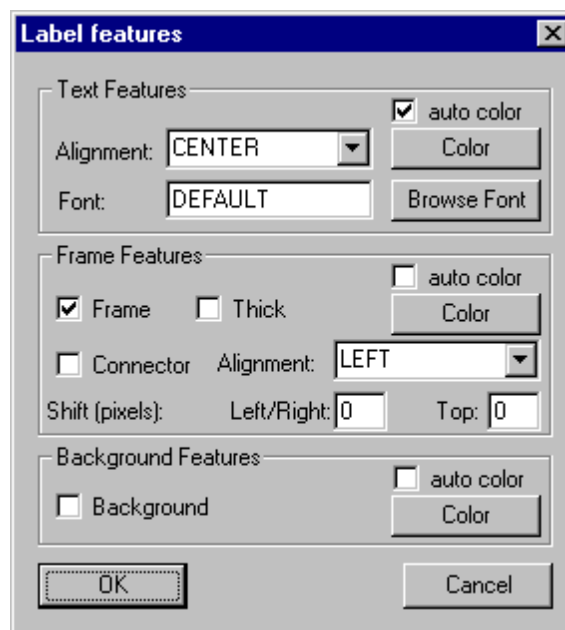
If not, a window will open that asks if you want to create a new network. If the answer is yes, a dialog box will open, where you can allocate a name to the new network file. Press the keyboard **Space Bar** to activate the editor. The cursor shape changes to a black arrow.

3. **Item name field** – Displays the name of the selected point or segment
4. **Change** and **Search** works only with points
5. **Change button** – Is only activated when a point is selected. Changes the name of a point to the new one typed in the **Item name** field. If the point name has 1, 2 or 5 letters the shape of the point will be a triangle, and the point can be moved. If the point name has 3 letters the shape is a star and the position of the point is locked.
6. **Search button** – Searches for point name typed in the **Item name** field. When found the point will be highlighted and centred.
7. **Lat/Long fields** – Displays the coordinates in Lat/Long of the selected point
  - a. **Change button** – Changes the coordinates of a point to the new ones typed in the **Lat/Long** field. The changed point will automatically move on the map.
8. **CDR check-box** – Changes a route segment into a CDR. Select a segment and tick the

- CDR check-box and then click outside the dialogbox, the segment will be displayed as a dotted line. This indicates that it's a CDR. The button 'CDR setting' is not yet implemented
9. **Parity drop-down list** – Options to change colour of a segment
    - a. **ODD** (green)
    - b. **EVEN** (blue)
  10. **Type drop-down list** – Options to change type of a segment
    - a. **NORMAL** (grey)
    - b. **ARRIVAL** (pink)
    - c. **DEPART.** (gold)
  11. **Size fields** – Changes the size of these objects
    - a. **Point** - Size of the points
    - b. **Seg.** – Thickness of the segments
    - c. **Arrow** - Size of the arrows
  12. **Labels frame:**
    - a. **Point Name check-box** – Displays point labels  
Button - Opens a new dialog box were you can customise the point labels ( see below)
    - b. **Nb seg Converge / Diverg fields** – An option to display just points with a certain number of segments converging or diverging to them
    - c. **Load Value check-box** – Displays the segment load  
Button - Opens a new dialog box were you can customise the segment labels ( see below)
    - d. **Nb Decimals. field** – Number of decimals displayed in segment load.
    - e. **Max. / Min. load fields** – An option to display just segments with a load within a certain range.

## Label Features

All labels can be viewed in 3D. They will always be facing forward irrespectively of how the display is turned.



**Label Features dialog box**

## Parameters

### 1) Text Features:

- a) **Alignment drop-down list** – Gives you three options were to display the text within the frame.  
**LEFT - CENTER - RIGHT**
- b) **auto color check-box** – Displays the text in same colour as the segment. Not implemented for points.
- c) **Color button** – To change the colour of the text
- d) **Font field** – A possibility to change the font of the text. Default font is **Arial**. By clicking on '**Browse Font**' you can select other fonts ( see below).

### 2) Frame Features:

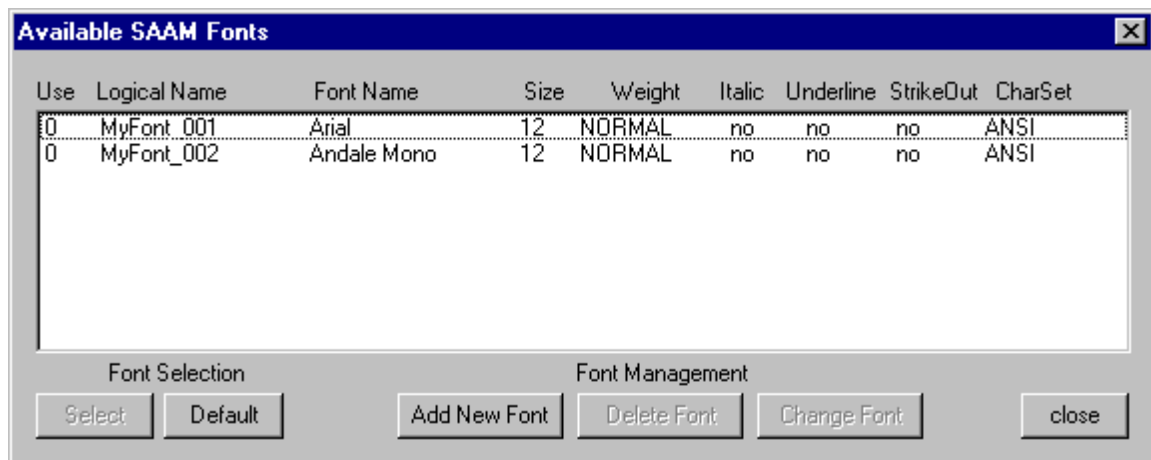
- a) **Frame check-box** – Displays a frame around the text
- b) **Thick check-box** – Makes the frame lines thick
- c) **auto color check-box** – Displays the frame in same colour as the segment. Not implemented for points.
- d) **Color button** – To change the colour of the frame
- e) **Connector check-box** – Displays a connecting line between the point and the label
- f) **Alignment drop-down list** – Changes the labels position in relation to the point.  
**CENTER - LEFT - RIGHT**
- g) **Shift (pixels) fields** – Changes the distance the label is positioned from the point.  
**Left/Right - Top**

### 3) Background Features:

- a) **Background check-box** – Displays a background
- b) **auto color check-box** – Displays the background in same colour as the segment. Not implemented for points.
- c) **Color button** – To change the colour of the background

## Change Text Font in Labels

If you want to change the text font in a point label you first have to click on '**Browse Font**' in the '**Label Features**' dialog box and a new dialog box will open.



Available SAAM Fonts dialog box

## Parameters

1) After have done ‘**Add New Font**’ (see below) a list will appear with the added fonts and its characteristics. The number under **Use** shows how many labels are using that font

2) **Font Selection buttons** –

**Select** – Selects the highlighted font from the list.

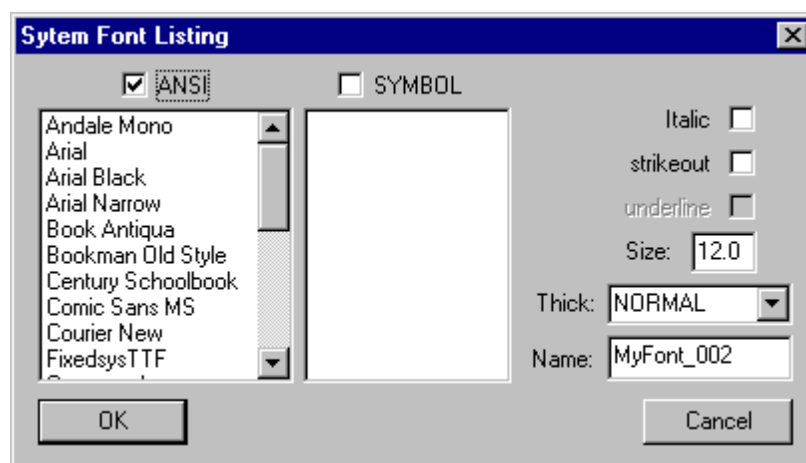
**Default** – Selects the default font, which is **Arial**.

3) **Font Management buttons** –

**Add New Font** - Opens a new dialog box (depicted below) with a list of different fonts.

**Delete Font** – Deletes the highlighted font from the list

**Change Font** – Opens a new dialog box (depicted below) were you can do the changes.



**System Font Listing dialog box**

## Parameters

1. There is a selection of fonts in ANSI and SYMBOL. When either check-box is ticked a list will be displayed, were you can select a font.
2. You can customise the font by choosing Italic and strikeout. Underline is not yet implemented. The thickness of the font can be changed by selecting one of the options under ‘Thick’.
3. When the selection and modifications of the font is completed you have to give it a name. You can either accept the suggested name displayed in the ‘Name’ field or type in a new one.

## Network Edit Operations

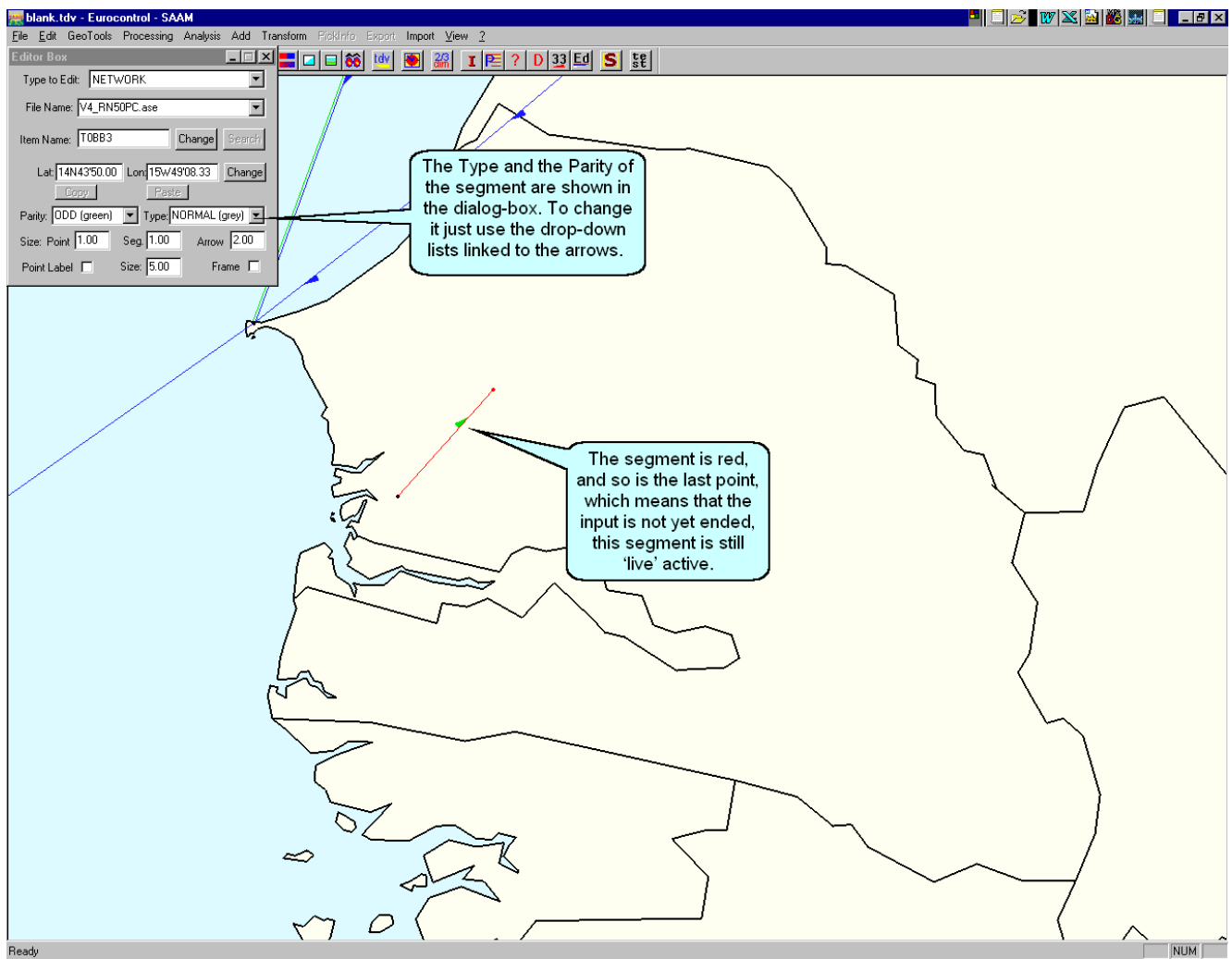
The following Network Edit operations are available

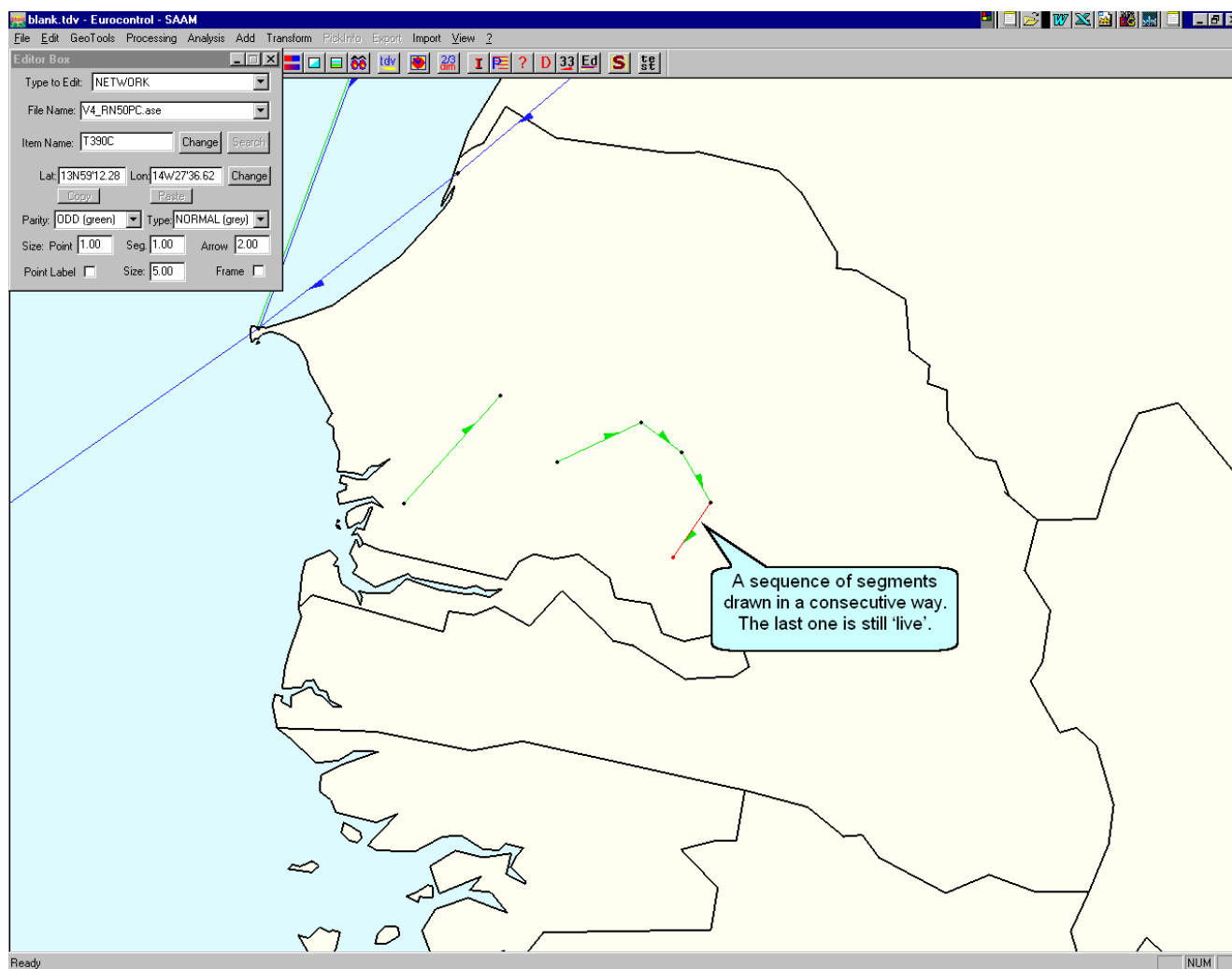


- |                  |                                     |
|------------------|-------------------------------------|
| • Select Segment | Left Mouse Click (LMC)              |
| • Create Segment | Ctrl + LMC, do this operation twice |
| • Delete Segment | LMC + Backspace ( ← )               |
| • Add Point      | Ctrl + LMC in an existing segment   |
| • Delete Point   | LMC + Backspace or                  |
|                  | LMC + X + Backspace                 |
| • Move Point     | LMC + drag                          |
| • Merge Points   | LMC + drag together                 |

## Create Segment

In order to create a segment, use Ctrl + LMC to create the first point, then Ctrl + LMC again to create the second point of the segment. The segment is then automatically depicted. By repeating this operation several times one can draw consecutive segments.





In order to create new segments which are not linked consecutively to the previous one(s), release Ctrl and click anywhere on the map. The last created segment will then change its colour from red to the chosen parity or type : it becomes 'muted'

To create a **bi-directional** segment, create the first segment Point A – Point B in the usual way, then click somewhere on the map to mute the segment. Use Ctrl-LMC to start another segment input but starting on Point B and then Point A.

## Delete Segment

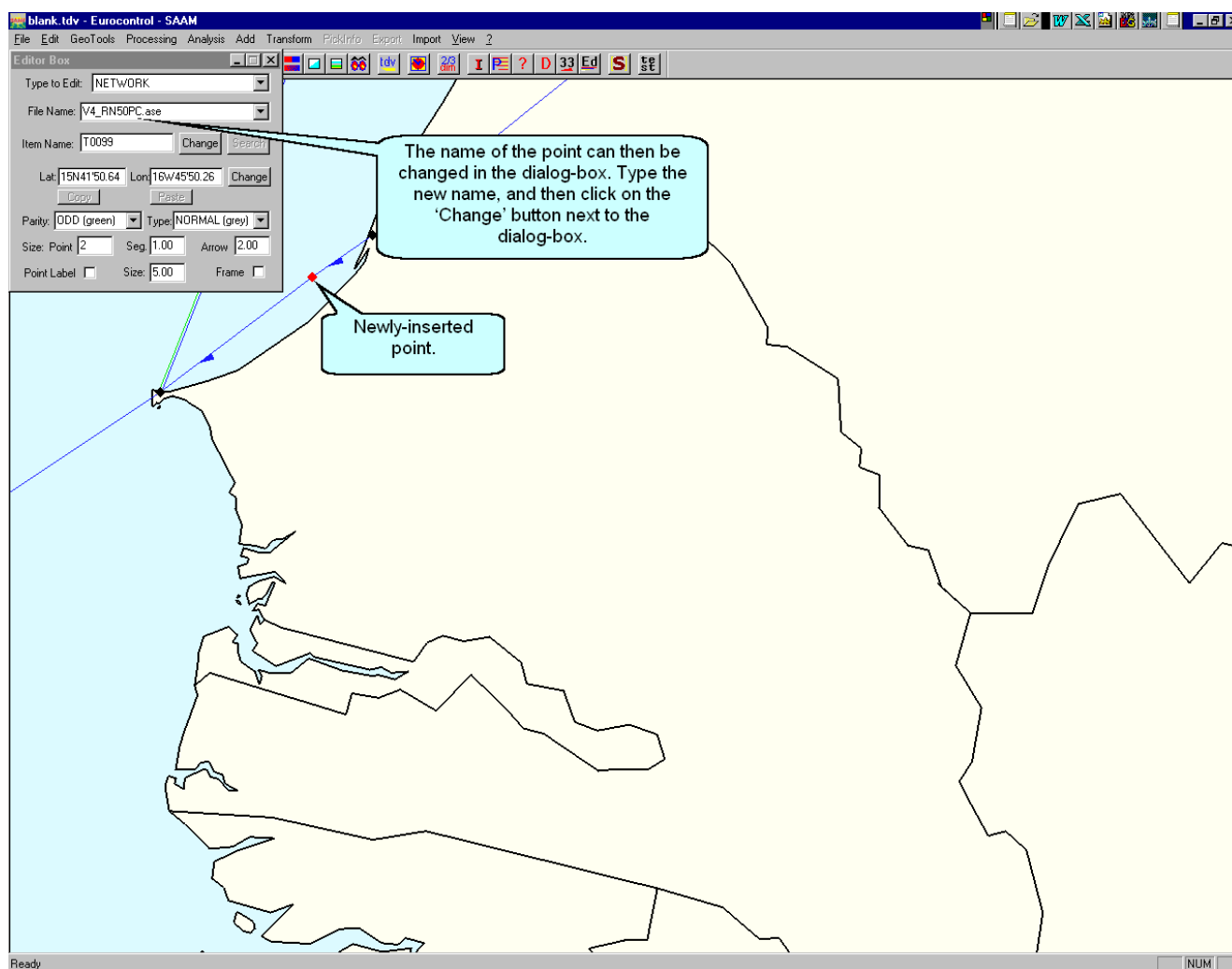
To delete a segment click on it and it will become red, indicating its 'live' status, then press 'Backspace'.

## Add Point

Points can only be inserted in an existing segment. Use Alt + LMC to insert a point in a segment. The point will by default be aligned to the segment and possible to move along the segment while the mouse button is not released.

Points are displayed with different shapes. A three letter point is displayed with a star. All

other points with a diamond. If the point is un-locked, can be moved, it is displayed with a triangle.



## Delete Point

There are two ways to delete a point: The first method just deletes the point but re-creates a segment between the previous and the next points. To perform this, click on the point to be deleted and use 'Backspace'. The segment will be automatically recreated without the point in it's middle. If the point to be deleted is between more than 2 segments this won't work.

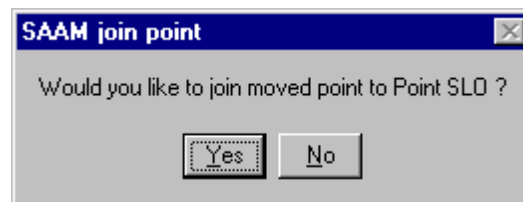
The second method deletes the point **and** the segments which are linked to this point. This is the only method that works if the point links more than two segments : click on the point, then press the 'X' key simultaneously with 'Backspace' : the point and ALL the connecting segments will then be deleted.

## Move Point

By default all points are locked. To move a point first you have to un-lock it. Select the point you want to move, un-check the check-box "Locked". Then click on the point and drag it, or input directly the coordinates in the relevant dialog-boxes and click then on the 'Change' button next to it.

## Merge Points

If you want to merge two points, drag the point to be merged on top of the other point. Release the mouse button. A dialog-box will appear with the following message:



*Clicking Yes will merge the two points together.*

## Lock point

Newly inserted or created points are UNLOCKED, so that users can work with them. The points will stay unlocked until the network is saved and re-loaded. Re-using a point, during a segment creation, will keep its lock status. As point lock status are not saved, all points will be LOCKED when network is re-loaded. User's still have the ability to manually lock or unlock points, by ticking or un-ticking 'Locked' check-box in the dialog box.



## Copy/Paste Segment

It is possible to copy/paste a selected segment from one SAAM to another or within same SAAM but from different networks.

Copy/Paste must always be done from the Route Network editor i.e. the Route Network editor must be open. Also when pasting, the network you want to past in must be selected.

Name of ending points are checked if existing or not. If they are found, the distance between current location and pasted location, for each of the two extremities, is measured, below 0.001NM it is assumed to be correctly located. Above this limit a dialog box opens, asking the user which location should to be used.

If the segment already exist, user is warned that pasted segment features, like parity, type , CDR ..., will overwrite current segment.

All pasted segment features (like parity, type...) are always pasted and possibly overwriting already existing segment. In such case the user is warned by a dialog box.

The segment load on a copied/pasted segment is always reset to zero.

## 4.3 SAAM Graph module

The SAAM Graph module allows you to display and print graphs of sector loads generated by the [Airspace Load](#), the [Controller Workload](#) and the [Combine Airspace curves](#).

The following functions are available:

- [opening a graph](#)
- [graph properties](#)
- [browse the graphs](#) on the screen,
- customise the [series properties](#)
- customise the [axis properties](#)
- [filter](#) to display a subset of the sectors present in the input data
- [copy/paste](#) a graph to Microsoft Office application
- [night view](#) to center the 24h horizontal axis around midnight
- [page layout](#) to change the general parameter of your prints
- [print graphs](#)

### 4.3.1 Opening a graph

A graph can be opened in 3 ways:

- manually via the **File → Open Graph** menu item to open an existing graph file. The same feature is available from within the SAAM main menu.
- automatically, when SAAM generates the graph data in the [Airspace Load](#), the [Controller Workload](#) or in the [Combine Airspace curves](#) function.
- by selecting one of the 4 last opened graphs listed in the **File** menu of the SAAM Graph module.

### 4.3.2 Graph properties

The SAAM graph displays one graph at a time in the main window.

The vertical axis represents a *Number of flights/aircraft*, a *Capacity*, the *Time spend in a sector (in minutes)* or a *Percentage variation between curves*

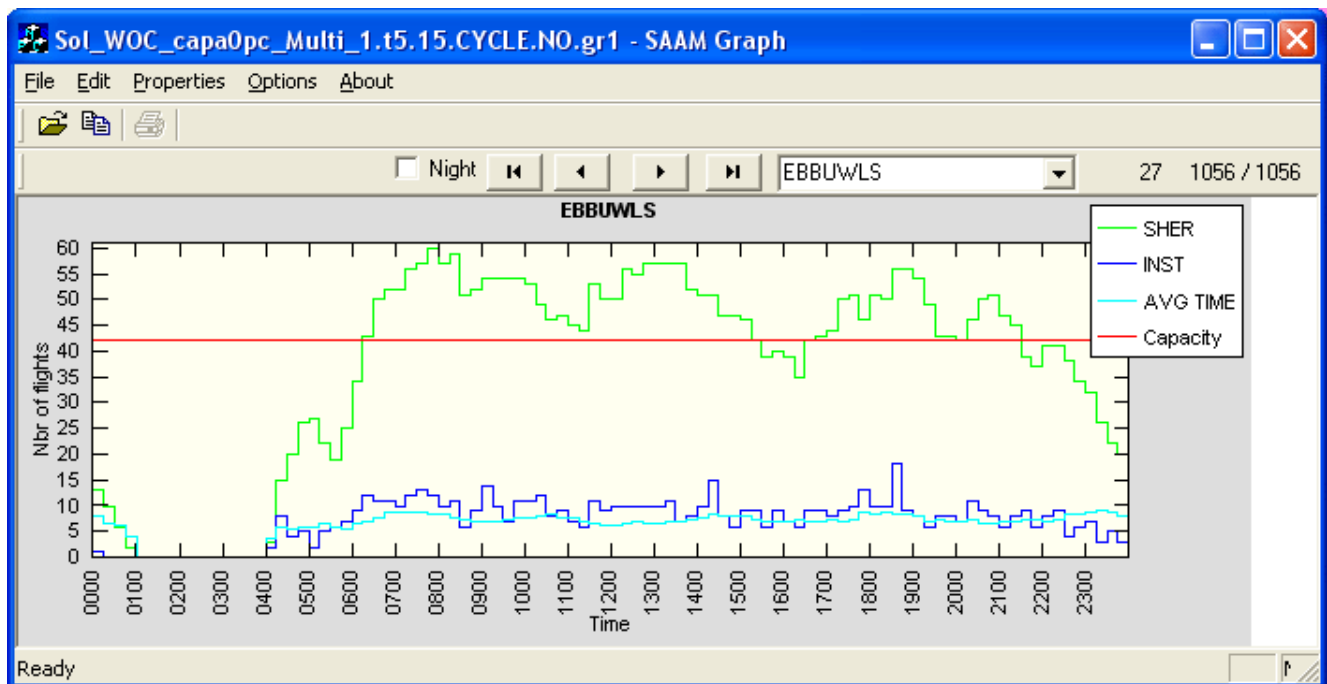
The horizontal axis shows the time of the day from 00:00 till 24:00. By default the time axis is centred around 12:00 but it can be changed to midnight (00:00) for a [Night View](#).

A graph contains one or more data series for each sector (eg SHER, INST, AVG TIME series). The [series properties](#) and [axis properties](#) can be customised.

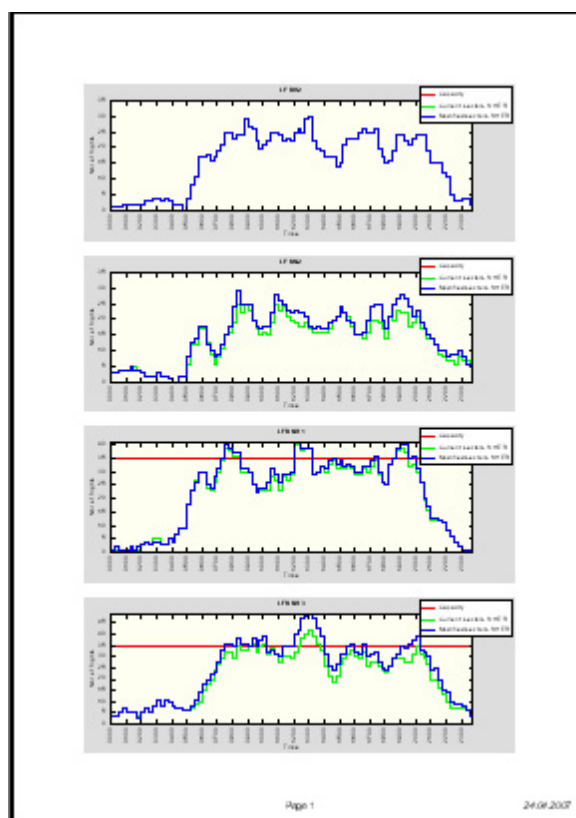
The number of points (steps) in a serie depends on the parameter you have selected in the SAAM analysis tool. By default it is 15 minutes, hence you get  $24 \times 4 = 96$  values per serie. If you change it to a 1 minute step e.g. to calculate the sector load every minute (via the [Time-slice duration](#)), the graph will contain  $24 \times 60 = 1440$  points.

The data for the curves are stored in [gr1 files](#). Note: you don't edit gr1 files since they are produced automatically by a SAAM analysis tools.

If you want to restrict the sectors displayed/printed, you can do it via the [sector filter](#).



**Main window of the SAAM Graph - browsing through the sectors**



**Sector loads printed on A4 paper**

### 4.3.3 Scrolling through graphs

The SAAM graph displays data series for one sector at a time. You can scroll through the sectors with the Windows navigation keys or via the application navigation toolbar.

The graphs appears in the same order as in the gr1 file. To navigate according to the name of the sector, use the Drop down list in the navigation toolbar. (see below)

#### Navigation keys (follows the gr1 sector order)

- **Up arrow** or **PgUp** : previous graph
- **Down arrow** or **PgDn**: next graph
- **Home**: first graph
- **End**: last graph

#### Navigation toolbar (follows the gr1 sector order)

Same as navigation keys.



#### Navigation via the drop down list (follows the alphabetical order of the sector names)



The sector drop down list is ordered alphabetically. If the drop down list is selected (the sector name has a blue background), you can go use the **Up arrow** or **Down arrow** keys to navigate in alphabetical order.

In addition you can access directly a specific sector by selecting it in the drop down list.

#### Sector numbers

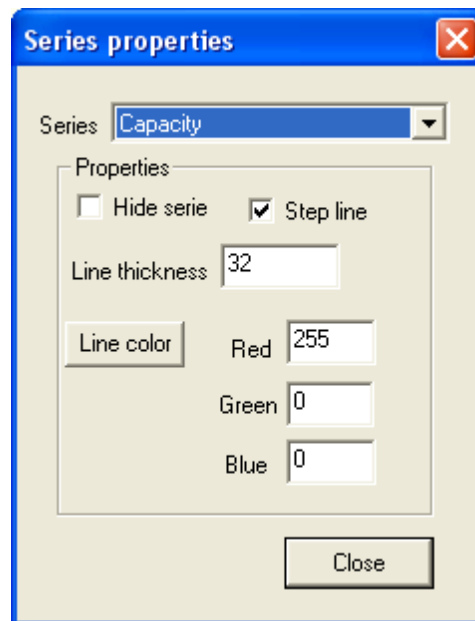


You can see the index number of the current sector and the total number of sectors in the navigation toolbar next to the drop down sector list. The first number (4 in this example) indicates that LFBNH4 is the 4th sector in the gr1 file. The second number (12 in this example) indicates the total number of active sectors ie the sectors that match the [filter criteria](#). The last number (15 in this example) gives you the total number of sectors in the gr1 file. The number of active sectors and the total number of sectors are the same if no filter is active.



### 4.3.4 Series properties

The graph series (e.g. SHER, Capacity, ...) have graphical properties (colour, ...) that you can change via the **Series Properties** menu item located in the main menu.

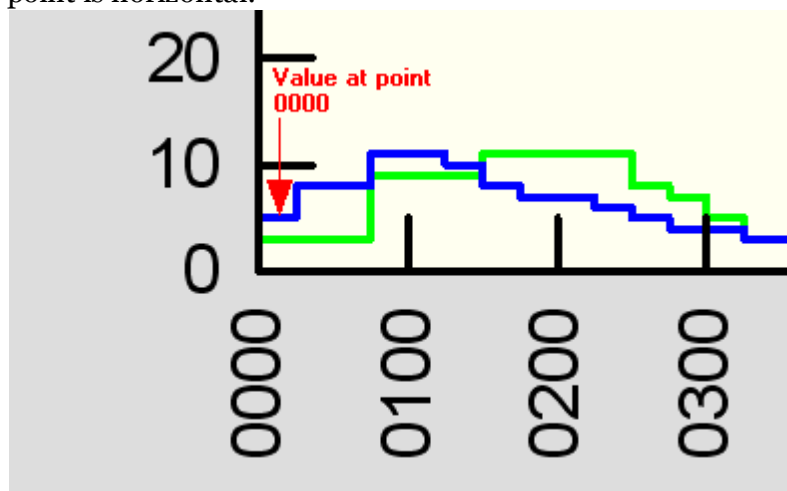


First select the serie you want to change in the **Series** drop down list and then modify its properties:

- **Hide**/Show the serie
- Tick **Step line** to plot data in steps (step line is the default value). If unchecked, the lines are plotted oblique
- Modify the **thickness** of the graph line
- Modify the colour of the graph lines. Enter a RGB value or select a colour in the colour picker window which can be called by clicking on the **Line color** button.

**Note:** your changes are not recorded with your graph file. They are lost when you close the SAAM graph module or when you open another graph.

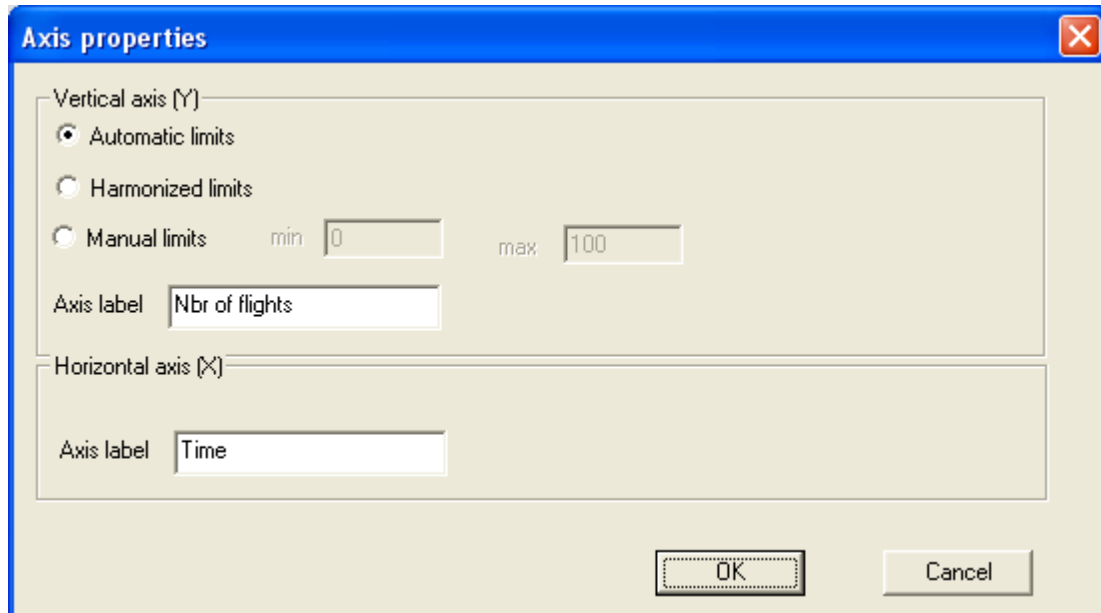
**Note:** the step line of the SAAM graph is of type horizontal: the initial line for each data point is horizontal:



### 4.3.5 Axis properties

The horizontal and vertical axis of the graphs can be customized.

To access the **Axis Properties** dialog box, go to the **Properties** item in the main menu.



You can change the scale of the vertical Y axis in three ways:

- **Automatic limits:** the scale changes for each graph. The maximum of the Y axis is equal to the highest value of the series over 24 hours (Warning: non visible series are also taken into account for this calculation). The minimum value is 0.
- **Harmonized limits:** the Y scale is the same for all graphs. The maximum value of the Y axis is equal to the highest value of the series of all graphs over 24 hours. (Warning: non visible series are also taken into account for this calculation). The minimum value is 0.
- **Manual limits:** you fix the scale of the Y axis manually by assigning min and max value.

The X and Y **Axis label** fields allows you to change the default labels generated by the SAAM analysis functions.

**Note:** you cannot define individual properties for each graph: the axis properties are common to all graphs in your gr1 file.

**Note:** the changes you make to the properties are not recorded in your graph file. They are lost when you close the SAAM graph module or when you open another graph.

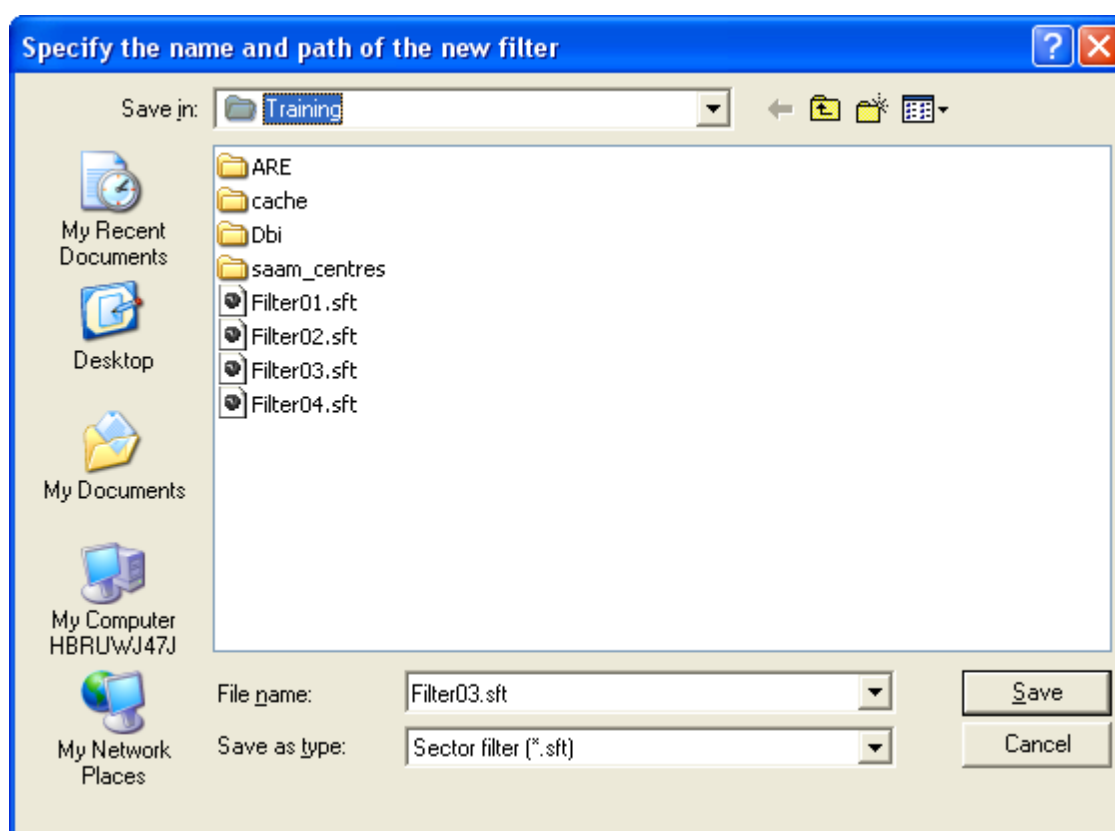
### 4.3.6 Filter

With the filter function, you can limit the sectors' graph displayed to a subset of the sectors present in your input file. To achieve this, you need to list in a text file ([.sft](#) file) the sectors you want to view.

#### Creating a new filter

Select **New Filter** in the File menu. A dialog box appears with a default filter file whose name is Filterxx.sft. If you want to rename it, go in the list of files, select the file to be renamed, click on F2 and give it a new name, then select it again so that it appears in the File Name box below.

Click on the **Save** button to close this dialog box. SAAM launches the SAAM text editor: start adding the sectors you want to keep (see Editing a filter, below)



#### Loading an existing filter

Select **Load Filter** in the File menu or click CTRL-F.

#### Editing a filter

Enter the name of the sectors you want to view, one sector per line.

Partial sector name are accepted: the filter will include all sectors beginning with the partial name. E.g. EB will retain all Belgian sectors. You can add comments by preceding text with the # sign.

Example:

```
# my list of sectors to display
ELFBN2 # Bordeaux sector
LFBX
EB # All Belgian sectors
```

## Editing / Activating-Deactivating the current filter

The main toolbar contains a **Edit** button to edit the current loaded filter and a check box to enable/disable the filter.



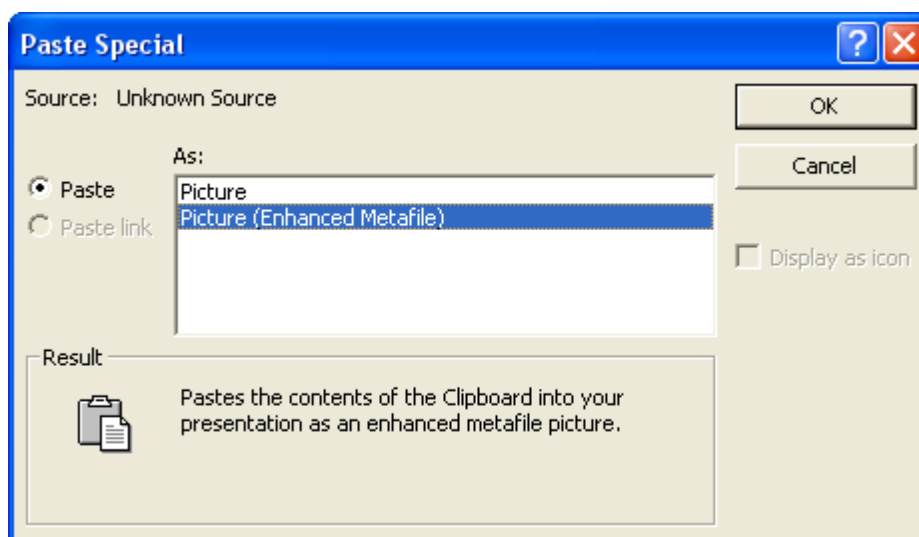
### Notes:

1. SAAM graph will not load an empty filter file (a file with no sector name)
2. If you filter excludes all sectors, the active sector count will be 0 and the sector drop down list will be empty but you will still see the last selected sector in the graph area.

### 4.3.7 Copy Paste a graph

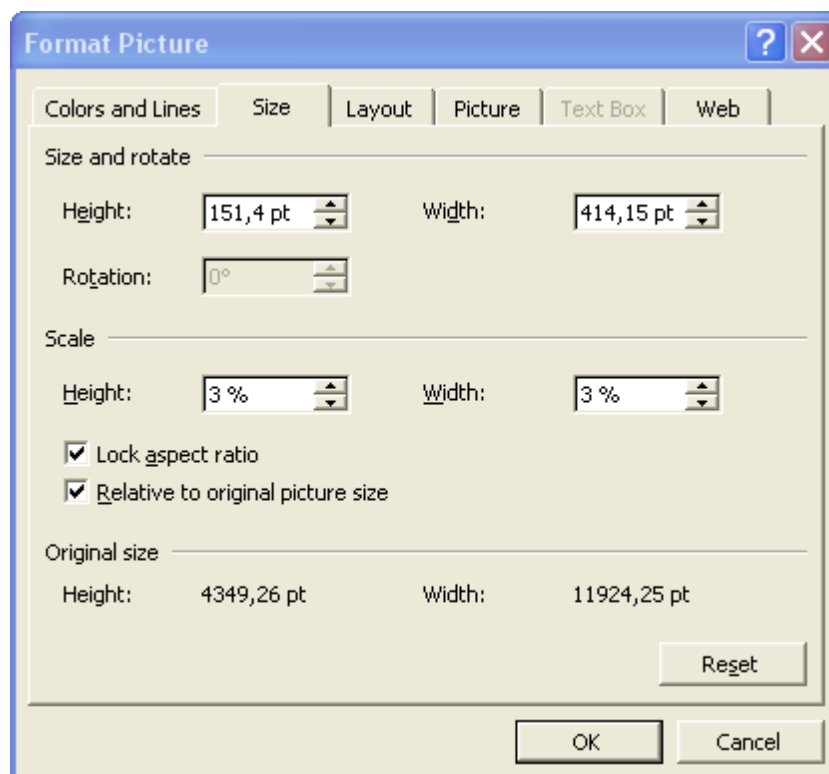
To export a graph into Microsoft Office applications (Word, Excel, Powerpoint, Publisher), use the copy/"paste special" functions:

- In the SAAM graph module, display the graph you want to copy
- Go to **Edit** -> **Copy** or type **CTRL-C**.
- Open the target application and select **Paste special** (in the Edit menu), select the option **Picture (Enhanced metafile)**. Note: if you don't see a **Paste special** in the Edit menu, click on the double arrow at the bottom of the menu to show all commands. Alternatively simply use the shortcut **Alt+E+S**



**Paste special dialog box in Microsoft Word**

- Resize the picture: use your mouse or use the Format Dialog box of the Office application:



Format picture in Microsoft Word

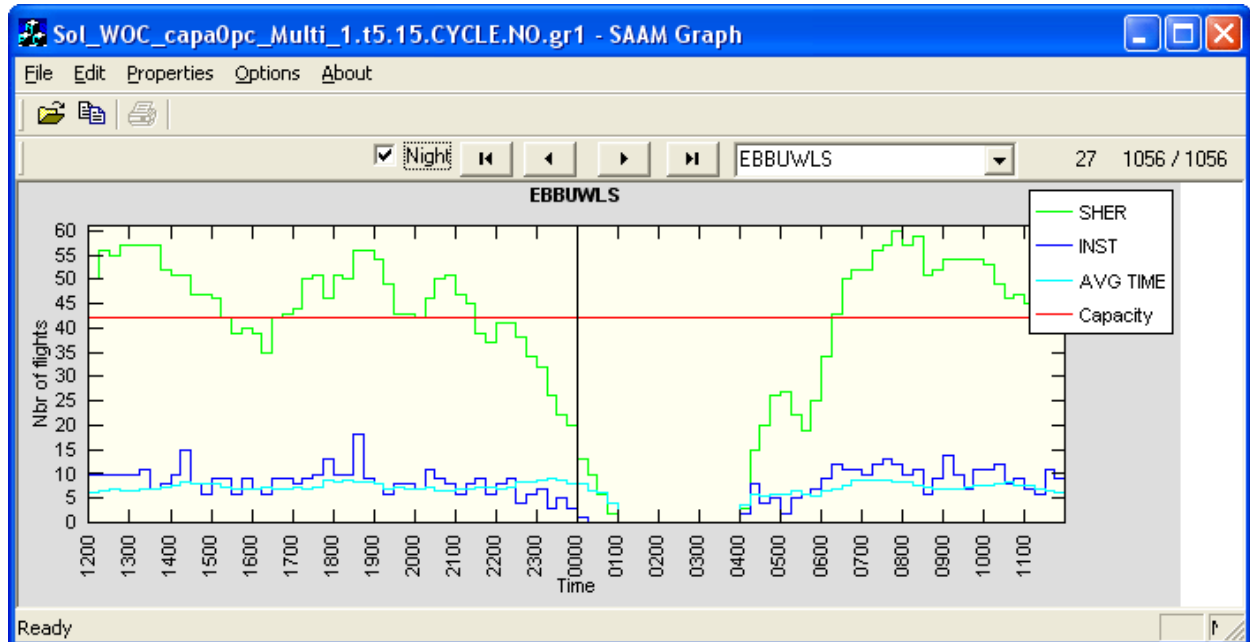
**Warning:** A SAAM graph can only be pasted correctly as a **Picture (Enhanced Metafile)**. If you use the default paste function (e.g. CTRL-V), some applications might use the bitmap Picture format of the clipboard content and the labels of the horizontal and vertical axis will be misplaced.

**Hint:** if your target application does not support the Enhanced Metafile format, you will need to convert the clipboard content to a bitmap format before pasting it.

**Note:** Since the SAAM graphs are defined with vectors (lines) and not in bitmap, you can increase or decrease its size in the target application without losing precision.

### 4.3.8 Night view

The Night view allows to center the graph around midnight. This is useful when you need to concentrate on the night traffic.



**Night view**

To activate the Night view, tick the **Night** box in the tool bar.

**Note:** the night View only moves the 00:00-12:00 data to the right of the graph and the 12:00-23:45 to the left. In other words, the data displayed always belong to the **same** day.

### 4.3.9 Page layout


#### Overview

You can decide how you graph will be printed on paper via the **Page Setup** menu and the **Printer settings**, both accessible from the File menu:

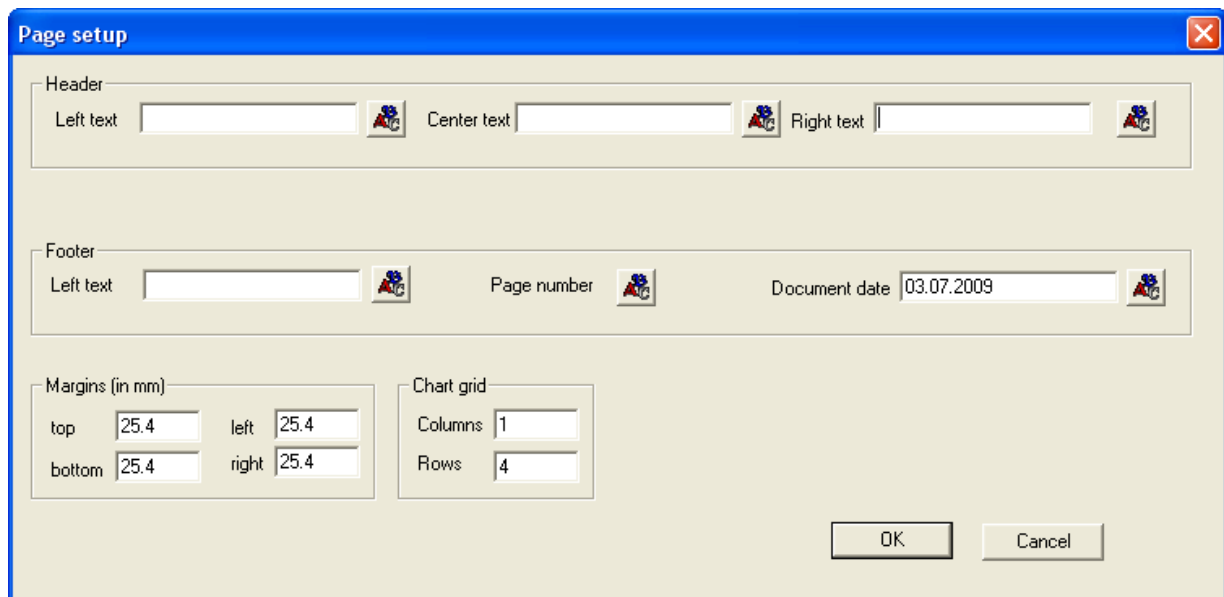
#### Page setup

To access this function, go in the **File** menu and select **Page setup**.

The Page setup allows you to specify a header, a footer, and margins for the print-outs of the graphs. The footer always include in its centre a page number and a date on the right (by default it is the current date).

You can change the font by clicking on the font icon .

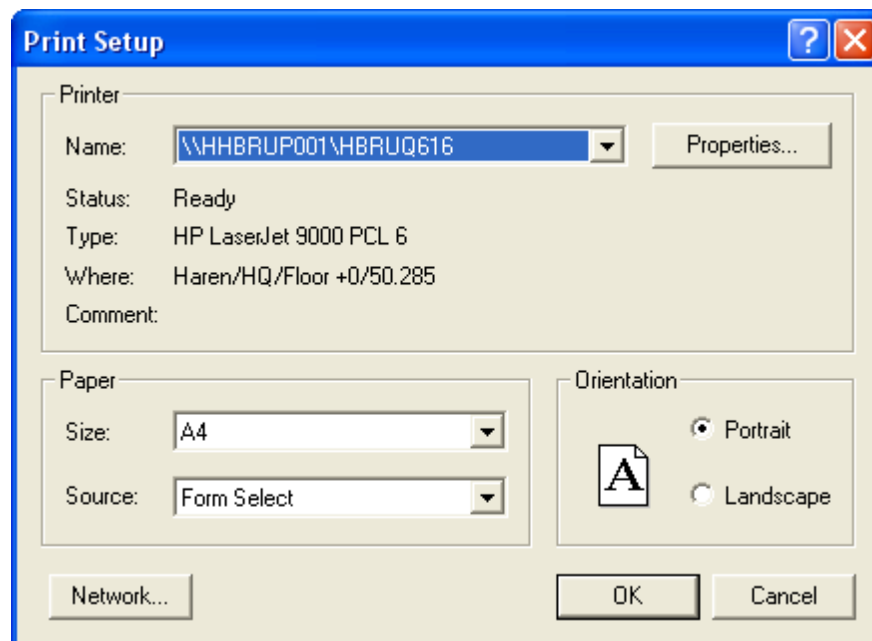
You can specify the number of graphs on a page, horizontally and vertically: **Graph Grid - Columns - Rows**. By default it is 4 graphs per page in one column.



## Printer settings

To access this function, go in the **File** menu and select **Printer settings...**

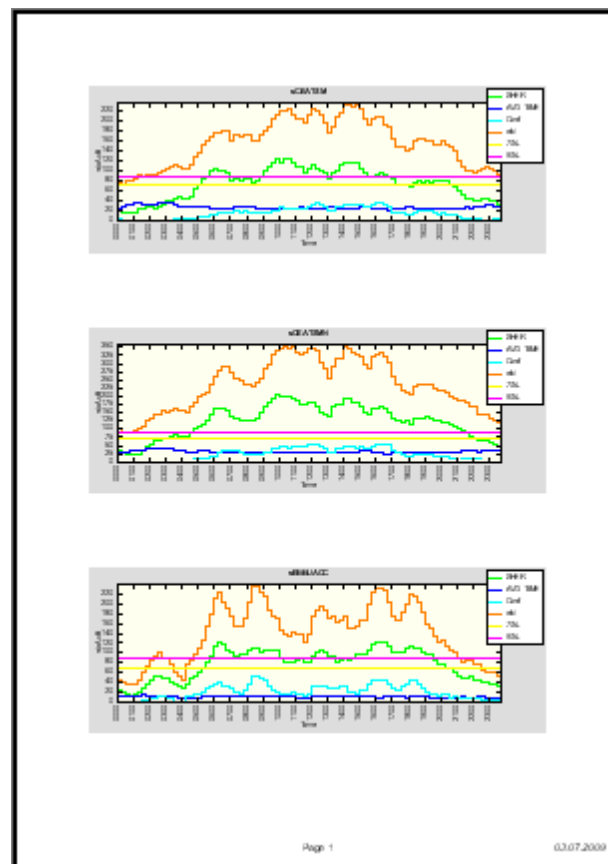
The **Print Setup** window allows you to select the paper **Size**, the **Orientation** (Portrait/Landscape) and the target **Printer**.



### Printer settings

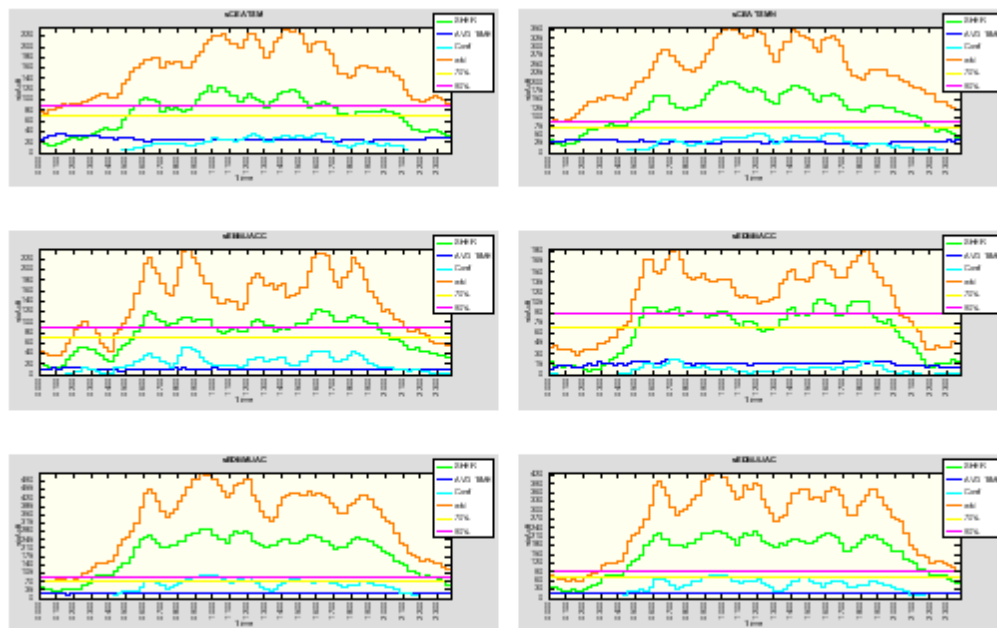
**Note:** the page layout is not recorded (in a file) and will be lost when you close the SAAM graph module or open another graph.

## Examples:



1 column, 3 rows





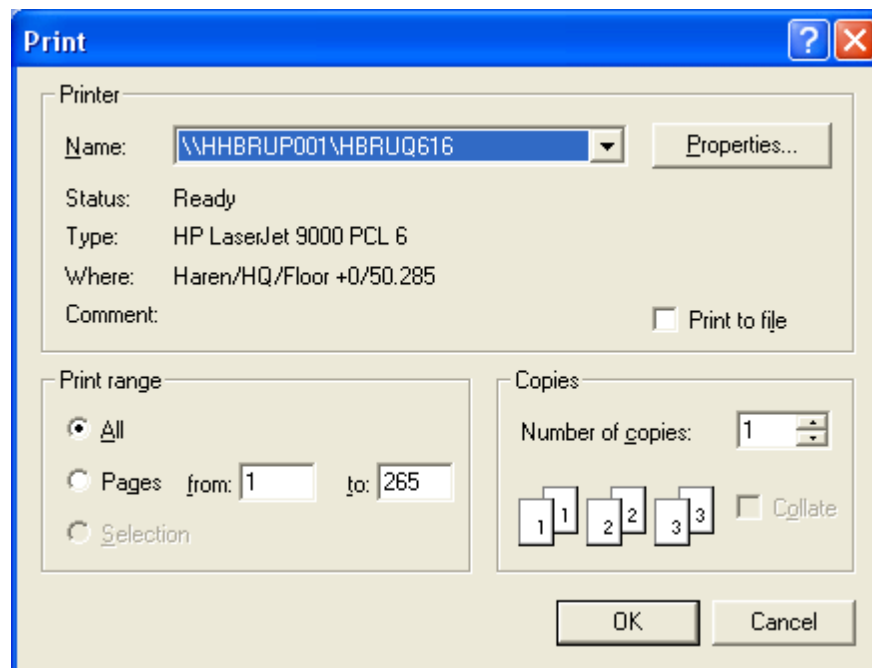
Page 1

03.07.2000

2 columns, 3 rows, landscape

### 4.3.10 Printing graphs

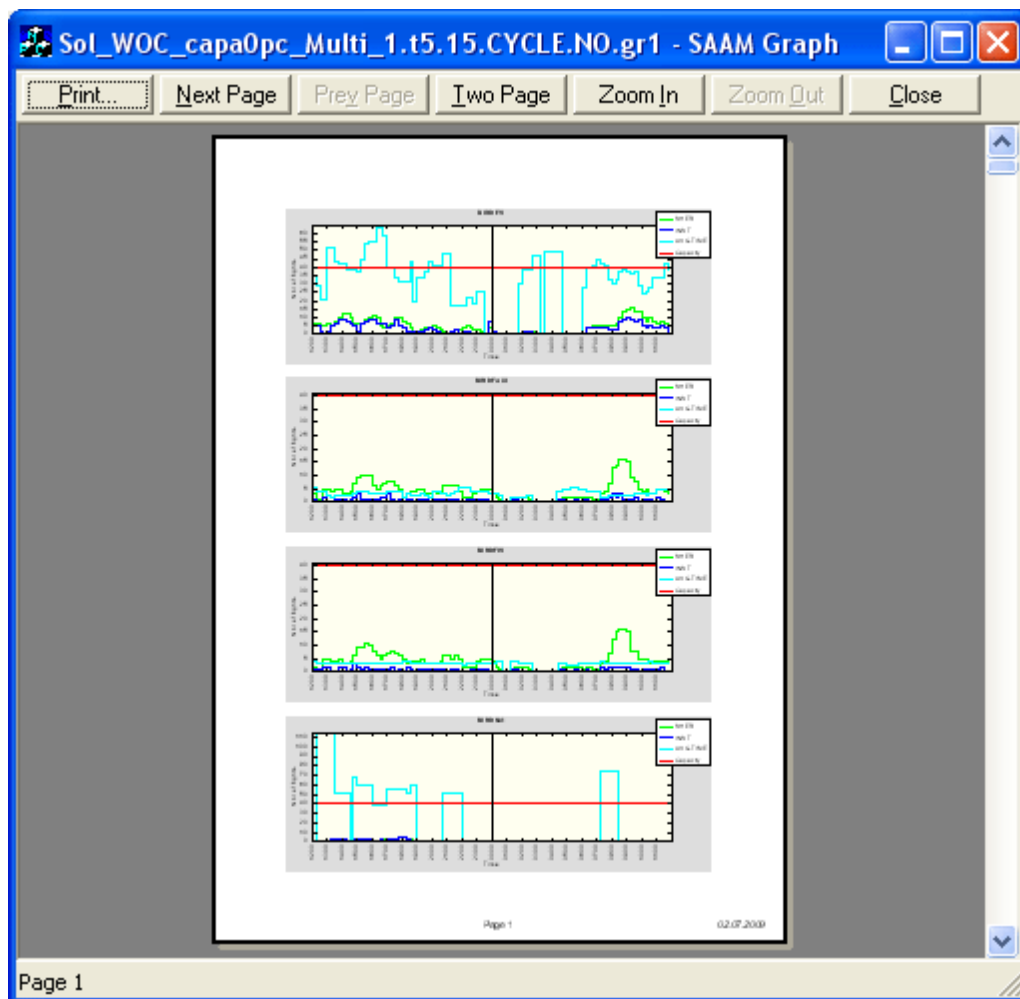
To print your graphs, select **Print** in the File menu. All pages are printed unless you specify a **from: to:** in the **Print Range:**




Saam graph print menu

Note: the layout of the pages can be customised with [Page layout](#) function.

Tip: before printing, make sure that the layout is correct with the **Print preview** function. You will save paper! Go to the **Print Preview** item of the File menu:



## 4.4 Using the TDV Editor

The TDV editor allows users to edit [Tdv files](#). You open it by clicking on the Tdv icon  in the toolbar

```
test_3.tdv - Notepad
File Edit Search Help
# SAAM TDV file
# User: lindbera
# Time: Tuesday, 07 October 2003 - 11:14:17
# Computer: NTW10031
# Name: blank.tdv
# Directory: J:\SAAM_Studies\6States\

BACK_COLOUR C 64 220 250 255
U S L EUROPE_COUNTRIES.ARE -6.00 -1.00 C 64 255 254 240

U T L ..\Data?files\020903\TMA_Total.are 999.00 999.00 C 128 255 0 0
NETWORK 6 LOAD ..\Data?files\020903\U4bis_24d.ase 200 100 50 20 10 0.50 1.00 0.80 0 0 0 C 255 255 0 0 1 X 0 C 0
XORIENT 0.000000 0.000000
YORIENT 0.000000 0.000000
ZOOM 0.000000 471453.156250
XMOVE 0.000000 0.000000
YMOVE 0.000000 0.000000
YUMOVE 0.000000 0.000000
AMORT 99999.000000 99999.000000 99999.000000 99999.000000
```

A text file will open containing several lines with information:

The first lines starting with # contains information about the user, the name of the TDV file

and its location.

The following line 'BACK\_COLOUR' indicates the colour used for the sea.

The next one indicates which background map is used. It can be either Europe or World map.

All of these lines should be left alone

## Adding and Editing an animation

Please refer to [Creating a SAAM animation](#).

## Editing Airspace Volumes

A line with airspace volume information will always start with 'V'. Explanations about the different fields will be found in the [TDV file format](#).

**V T T ARE\sEDYYLNO.are 999 999 C 80 117 116 242**

If you want to rename an airspace volume, you must change the name both in the TDV file and the \*.are file in its location.

After editing the file, **SAVE** it and then close the window. The picture will be regenerated. If you have forgotten to change the name of the \*.are file in the default directory you will get an error message:

To solve that problem, rename the file and then reload the picture.

## Editing Networks

The network display line is a little more complex: among other things it contains the limits for segment loads. It is advised to copy the entire line as it is in the example, modifying only the name of the network file.

**NETWORK 5 LOAD V4bis\_24d.ase 200 50 20 10 1 0.50 1.00 0.80 1 5.000000 0 0 0 C 255 255 0 0**

**NETWORK 5 DEFAULT V4bis\_24d.ase 200 100 50 20 10 0.50 1.00 0.80 0 0.000000 0 0 0 C 255 255 0 0**

Field 1 and 2 should always be 'NETWORK 5'. Field 3 indicates the default display type: MIXED, PARITY or LOAD (this can be toggled using Alt-N). Field 4 indicates the \*.ase file which is displayed. It has to be present at the indicated location otherwise SAAM won't work.

The next 5 fields indicate the limits for segment load colours. The last fields should better be left alone.

## Adding and Editing a Point, Line, Shape, Label, Title, Image

Please refer to [Adding graphical objects](#)

## Adding runway data

Please refer to [Add runway](#)

## Miscellaneous Editing

Additional items can be added to the TDV: see [TDV file format](#).

### 4.5 Modifying the Assignment & Profile Parameters

Traffic assignment relies on various parameters. All of these parameters are text files which can be edited by the user. However, great care should be exercised while doing this as faulty parameters or corrupt files will have a disastrous effect on the accuracy of the results.

#### Traffic Demand

This file is generated by the so6 to expand function. An expand file (\*.exp) contains the following infos:

Dep. Airport ; Arr. Airport ; (unused) ; Aircraft Type ; Requested Flight Level ; (unused) ; (unused) ; # of flight ; Date of Dep. ; Hour of Dep. ; (unused) ; Callsign ; Company.

**Example:**

**ZBAA LFPG ? MD11 370 ?? 26703 020607 1925 ? CES5793 N50**

#### Using Airports or Zones

In order to reduce the number of data, airports can be grouped inside a Zone (ZOD). All airports that are grouped inside the same ZOD will follow the same Sids and Stars and the same 2d rules, unless they are to be found in the Sid, Star and 2d rule files.

Zones are defined via the AirportToZone file. This is an \*.awk-type file (see section 6 xxx for explanation on the awk language) which can be edited with a text editor. The syntax is:

**if (i ~ /LIPZ|LIPQ|LIPA|LIPH|LIPB|LIPU/) return « LIPZ »**

The first part of the expression contains the ICAO codes for the various airports which are part of the zone, while the second part is the name of the zone itself (= Head of zone)

The contents of the Sid, Star and 2d rule files will always have priority to the contents of the AirportToZone file. An airport that have specific Sid- and Star points defined to it will follow these, even though it's belonging to a zone. The same goes for airports in the 2d rule file.

### SID and STAR Points

SAAM does not need a physical link (i.e. a segment) between airports and the network. Instead, the software will look for the closest point on the network to the airport. It will then perform a short-haul hop from the airport to that point (or from that point to the airport if it is an arrival). This search is performed in the general direction of the flight, i.e. if the flight is westbound, SAAM will look for points west of the departing airport.

#### SIDs and STARS

The method mentioned above is satisfactory for small airports, but in the case of major airports, specific Sids and Stars are usually devised. If specific departure and arrival points are defined for an airport, the traffic will use only these points for entry or exit into the network,

and not this method. These points are stored in two files : xxx.sid and xxx.star

The syntax of these files is self-explanatory:

Name of Airport Point1 Point2 ...

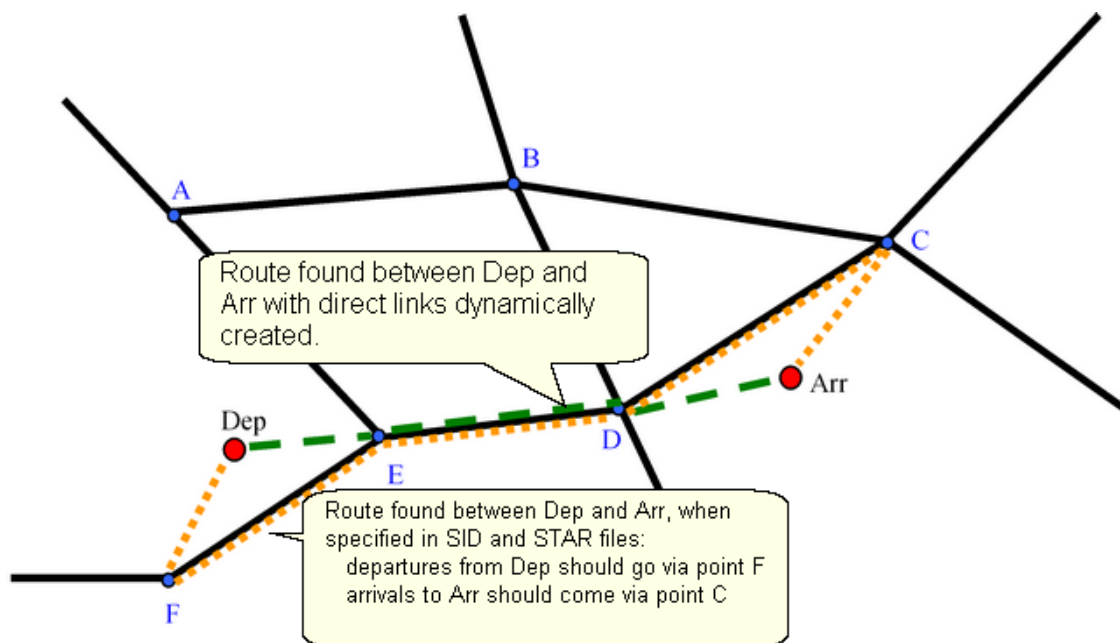
**EGCC MCT DENBY WAL NOKIN**

### A special case: the North Atlantic flights

Flights across the north Atlantic usually fly westbound on a more northern route than the orthodromy. This is due to the wind pattern in this region. In order to simulate this the SAAM Development Team have created a special case:

A special NATS zone has been created for which a large set of points have been labelled as Sids and Stars. Thus all the flights crossing the Atlantic will fly to or out of Europe through these points. However specific rules have been devised in order to shift part of the traffic from central Ireland (the shortest route) to Scotland (usually the preferred route). Thus in this case the 2d rules are disconnected from the ZOD definition.

### Example when using SID and STAR files:



## Flight Level Constraints

### 3D Constraints

In the same way that 2D rules aim at increasing the capacity by organising the traffic in 2D, 3D constraints are used for the same purpose but in 3D.

Two basic methods are used to enhance airspace capacity through 3d constraints:

1. Keep short-haul traffic lower than long-haul traffic
2. Decrease the number of sectors that an evolving flight is crossing

3d constraints are all within the contPinxxx.flc2 file which is also an awk-type text file. The syntax is similar to that which is used for 2D rules, except for the following point:

## CAUTION:

---

flc2 files are processed with full regular expression capabilities. In particular users should pay attention to the difference between == and ~ especially for 3 letters points case.

- Using == always gives exact, simple (and quick) matching (name must be between double quotes)

if (ARR ~ /EGSG|EGSS/ && **POINT=="NOR"** ) FL\_CONT=-130 // **NO PROBLEM the point can ONLY be DTY**

- Using ~ gives extended capabilities (name, or list of names with logical operator and/or particular delimiters must be between //).

if (ARR ~ /EGSG|EGSS/ && **POINT~/NOR/** ) FL\_CONT=-130 // **POSSIBLE PROBLEM with point called NORPI or NORMI !!!!!!!**

Above expression POINT ~/NOR/ means all POINT having the name NOR within their names, so for instance NORMI or NORPI or ANORB will be matching, which is something that you might not want ! So for 3 letters point name we strongly recommend to use == instead of ~.

---

## En-Route Constraints

The basic cruising constraint syntax is as follows:

**if( DEP ~/LIM./ && ARR ~/LEBL|LEGE/ && \$3==0 ) FL\_CONT = 290**

**= city-pair Milano TMA to either Barcelona or Gerona may not fly above 290. The condition '\$3 == 0' indicates that this is a cruising level constraint.**

## Arrival and Departure Constraints

These constraints indicate both a max FL, which is authorised and a point at which this condition must be met.

**if( DEP ~/EB../ && POINT == « PON » ) FL\_CONT=+330**

**= all departures from Belgian airports via PON must cross PON at FL 330 or below.**

Practically it means that the aircraft will climb until FL 330 and then remain at that level until it reaches PON. Afterwards it may resume its climb.

An arrival constraint will be written as:

**if( ARR ~/EB../ && POINT == « VEMAK » ) FL\_CONT=-220**

**= arrivals to Belgian airports via VEMAK must cross VEMAK at FL 220 or below.**

Or :

```
if( ARR == « EBAW » ) {
    if( POINT == « FLORA » )    FL_CONT=-180
}
```

= arrivals to Antwerp via FLORA must cross FLORA at FL 180 or below.

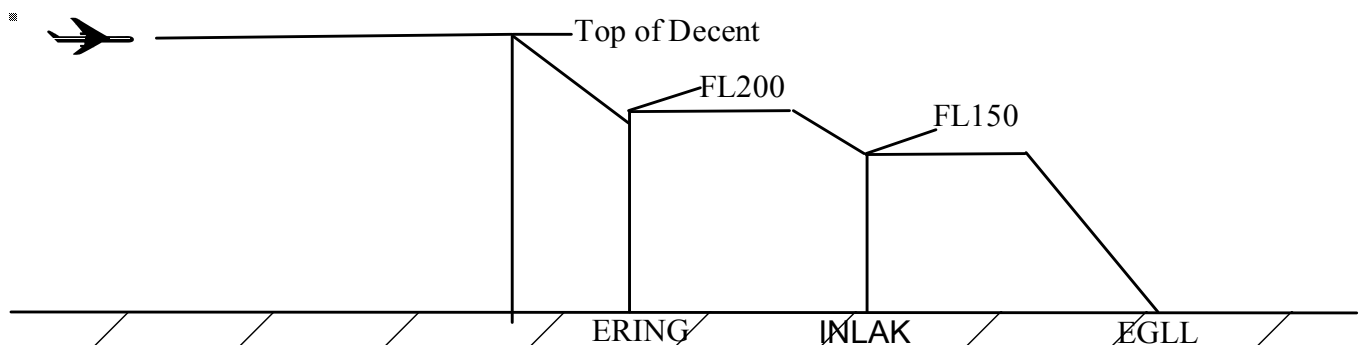
## !! BEWARE !!

A departure constraint will be coded as a '+' (plus) ; an arrival constraint as a '-' (minus). However both indicate a FL not to be exceeded.

Several constraints can be combined. This is very common for large airports.

```
if( ARR == « EGLL » ) {
if( POINT == « ERING » ) FL_CONT=-200
if( POINT == « INLAK » ) FL_CONT=-150
if( POINT == « HON » ) FL_CONT=-190
if( POINT == « LAM » ) FL_CONT=-110
if( POINT == « REFSO » ) FL_CONT=-240
}
```

Arrivals to Heathrow must be at the specified levels or below at the specified points: ERING 200, INLAK 150, etc some of these constraints can be consecutive i.e. on the same Star. This produces a stairway-type flight profile:



Also identical constraints can be used for various airports (this assumes of course that they share a common Sid and/or Star)

```
if(ARR ~ /EGSG|EGSS/ && POINT== »DTY » ) FL_CONT=-130
```

## CAUTION -

As with the 2D constraints, there is no debugging facility available yet and syntax errors will not be detected. Users are advised to input or modify data with an extreme care (beware of missing parentheses!)



## 4.6 Segment Management Rules

By default, the software assigns traffic on the shortest route that is available on the network. However, in order to mimic real life, one can specify restrictions for segments or flights. These are called Segment Management rules.

The rules are stored in \*.awk files and you can open it with txt editor (recommendation Notepad 2).

The file name is \*\_\*\_Segment\_Management.awk

The syntax for the rules is:

**if <Condition> <Rules>**

If the condition is met, then the rule is applied.

### CONDITION

**<Condition>** can be:

**DEP** - for Departure/Origin Airport

**(DEP ~ /EBBR/)**

**(DEP~ /Brussels/)**

with Brussels group defined as "EBAW EBBR EBCI EBCV EBMB" in the Area Definition file.

**ARR** - for Arrival/Destination Airport

**(ARR ~ /EBBR/)**

**CP** - City Pairs (Origin Destination)

**(CP ~ /EBBR LSZH/)**

**VIA** - for via Point

**(VIA ~ /ALURA/)**

examples:

**if (ARR ~ /LFPG|LFOB/ && VIA ~ /ROMTA|DERAK/) { REGLE=REGLE "-DJL" }**

Flights arriving at LFPG or LFOB and flying via ROMTA or DERAK are not allowed to fly via DJL. They can fly via DJL but in this case, they can not fly via ROMTA or DERAK. Or they can fly neither via DGL, nor via ROMTA or DERAK. The shortest solution will be kept.

**VIA** - for via Segment

**(VIA ~ /RETNO\_LASUR/)**

examples:

```
if (VIA=="RETNO_LASUR") {REGLE=REGLE " - LASUR ETREK"}
```

To forbid a specific turn at LASUR : Segment RETNO\_LASUR can not be followed by segment LASUR\_ETREK. This is equivalent to the rule :

```
if (VIA=="LASUR_ETREK") {REGLE=REGLE " - RETNO LASUR"}
```

**ACFT** - for Aircraft Type use ICAO aircraft type designators

**(ACFT ~ /B737/)**

```
if (DEP ~ /LKPR|LKVO|LKKB/ && (ACFT ~ /B737/)) { REGLE=REGLE " - LKD01 LKD03 - LKD02 MEDOV - LKD01 LKD10" }
```

**ACFT\_CAT** - Aircraft Category (Engine Type) Jet, Turboprop, Piston or Electric

The different aircraft categories can be found in a file called "actype.txt", located in bin directory of SAAM.

**( ACFT\_CAT ~ /Jet/)**

example:

```
if (DEP ~ /LKPR|LKVO|LKKB/ && (ACFT_CAT == "Jet")) { REGLE=REGLE " - LKD01 LKD03 - LKD02 MEDOV - LKD01 LKD10" }
```

```
if (DEP ~ /LKPR|LKVO|LKKB/ && (ACFT_CAT ~ /Turboprop|Piston|Electric/)) { REGLE=REGLE " - LKD02 LKD04 - LKD02 LKD13" }
```

**CALLSIGN** - Callsign

example:

```
if (CALLSIGN ~ /ABC123/) { REGLE=REGLE " - PT1 PT2" }
```

segment "PT1-PT2" is forbidden for flight ABC123.

```
if (CALLSIGN ~ /ABC..../) { REGLE=REGLE " - PT1 PT2" }
```

segment "PT1-PT2" is forbidden for company ABC.

**RFL** - Requested Flight Level

RFL from exp file Above FL 285

**(RFL > 285)**

RFL from exp Below FL 195

**(RFL < 195)**

example:

```
if (DEP ~ /EBBR|EBCI|EBLG/ && RFL < 245 || DEP ~ /EBAW/) { REGLE=REGLE "
- SOPOK ETENO" }
```

### Wildcard

The "." (Point) is used as wildcard ( to replace character within / / )

EB.. - all Belgium airports

### Combine condition with logical operators:

&& - means AND

|| - means OR

&&! - means AND NOT/Except (recommendation use space after !)

**if ( (Condition1) && (Condition2) )**

= condition 1 AND condition 2

**if ( (Condition1) || (Condition2) )**

= condition 1 OR condition 2

**if ( (Condition1) && !(Condition2) )**

= condition 1 Except condition 2

### Examples:

"|" means OR and is used in the definition of the Airports:

```
/LFJL|LFST|EDDS|LFSB/
```

= Louvigny OR Strasbourg OR Stuttgart OR Basel.

**if(DEP ~ /EBBR|EBLG|EBCI/) { REGLE=REGLE "- SPI WILEM" }**

= Flights departing from Brussels, Liège or Charleroi are not allowed via SPI-WILEM

**if (DEP~/LFST/ || DEP~/EDDS/) { REGLE=REGLE « + STR POGOL » }**

= STR-POGOL is reserved for departures from either Strasbourg or Stuttgart.

The double " | " means OR (Thus the syntax is different if the expression is inside the parentheses or outside)

**if (DEP ~ /LEPA|LEIB|LEMH/ && ARR ~ /LO..|LK..|EDDT/) { REGLE=REGLE « TORTU » }**

= Departures from the Balearics to Austria, Czech Republic and Berlin have to fly via TORTU

If an airport has a large set of rules, the following syntax can be used.

```
DEP ~ /EGKK/ {
if ( ARR ~ /C.../ ) REGLE= REGLE « LESTA »
if ( ARR ~ /KMSP/ ) REGLE= REGLE « E »
if ( ARR ~ /KMSP|KD..|KCLE|KSTL/ ) REGLE= REGLE « F »
if ( ARR ~ /KJFK|KEWR|KB..|KCVG|KPHL|KPIT|KCLT|KIAH/ ) REGLE= REGLE « G »
if ( ARR ~ /MMUN/ ) REGLE= REGLE « H »
if ( ARR ~ /K.../ ) REGLE= REGLE « - CHELT DUB - INLAK CPT »
}
```

The initial condition is departures from Gatwick; the sub-conditions are arrivals to various American and Canadian airports.

## RULES

### FORBIDDEN Rules

Forbidden Point restriction (no space after -)

```
if <Condition> { REGLE = REGLE "-PT" }
```

Forbidden Segment restriction (space after - and between Points)

```
if <Condition> { REGLE = REGLE " - PT1 PT2" }
```

example :

```
if ( CP ~ /EGNT EGGD/ ) { REGLE=REGLE « - DENBY LIC » }
= Newcastle to Cardiff is forbidden to fly through DENBY-LIC
```

### RESERVED Rules

Reserved Segment restriction (space after + and between Points)

```
if <Condition> { REGLE = REGLE " + PT1 PT2" }
```

example:

```
if ( CP ~ /LSZH EDDF/ ) { REGLE=REGLE « + KLO ZUN » }
= Only Zurich to Frankfurt are allowed on KLO-ZUN
```

Several rules can be combined in one line:

```
{ REGLE=REGLE « + ATN RESPO + DIJ OKRIX » }
=Reserved segments are ATN-RESPO and DIJ-OKRIX
```

**COMPULSORY Rules**

Compulsory (Mandatory) Point restriction (no space after >)

```
if <Condition> { REGLE = REGLE ">PT" }
```

example:

```
if (CP ~ /LFP. KSFO/) { REGLE= REGLE ">RONAR" }
```

= All flights from TMA Paris to San Francisco must fly via RONAR

Compulsory (Mandatory) Segment restriction (space after > and between Points)

```
if <Condition> { REGLE = REGLE "> PT1 PT2" }
```

example:

```
if (CP ~ /LFPG LIR./) { REGLE= REGLE "> KURIR OGALO " }
```

= All flights from Paris to Rome must fly via KURIR-OGALO

**Mixed Rules**

Different types of Rules (Mandatory/Reserved/Forbidden) can be mixed on the same line. Here is RESERVED and COMPULSORY rules mixed.

```
if <Condition> { REGLE = REGLE " +PT1 >PT1" }
```

**Tips and Tricks****RESERVED and COMPULSORY segment**

```
if <Condition> { REGLE = REGLE " + PT1 PT2 >PT1 PT2" }
```

Forbidden turn : use [Forbidden Turn Angles file](#)

**Compulsory turn**

```
if (VIA ~ /PT1_PT2/) { REGLE = REGLE "> PT2 PT3" }
```

**Reserved turn**

```
if (VIA ~ /PT1_PT2/) { REGLE = REGLE " + PT2 PT3" }
```

**Segment is not available to any traffic**

**if (DEP~/.../) {REGLE = REGLE " - MY SEGMENT"}**

**if (ARR~/.../) {REGLE = REGLE " - MY SEGMENT"}**

Typing four dots instead of airport name, means all airports. This can be used instead of a penalisation file, add 1000 extra NM to a segment, to make sure that no traffic will be assigned via a specific segment.

### Modification of the RESERVED rule into a FORBIDDEN rule

Restriction:

**Segment LENDO-NOR Only available for traffic Destination ED\*\* except Destination EDAB, EDAC, EDDB, EDDC, EDDE, EDDP, EDDT**

Instead of writing :

**IF (ARR~/ED../ && !(ARR~/EDAB|EDAC|EDDB|EDDC|EDDE|EDDP|EDDT)) {REGLE = REGLE " + LENDO NOR"}**

You write :

**IF (!(ARR~/ED../) || ARR~/EDAB|EDAC|EDDB|EDDC|EDDE|EDDP|EDDT)) {REGLE = REGLE " - LENDO NOR"}**

Second example :

if the following is specified in RAD :

**Segment DET-VABIK**

**Only available for traffic**

**Dep. EG\*\* (except Farnborough Group , EGDR/DY/HD/HH/HI/ HQ/HR/KA/KB/KK/LC/MC/MD/MH/SC/SS/TE/TO/VP)**

**With Dest. Brussels Group, EBLG, Paris Group, EH\*\***

then, it shall be reworded into the following for proper assignment functioning :

**if ((!(DEP~/EG../) || (DEP~/EGHL|EGLF|EGLK|EGTD|EGTF|EGVO|EGDR|EGDY|EGHD|EGHH|EGHI|EGHQ|EGHR|EGKA|EGKB|EGKK|EGLC|EGMC|EGMD|EGMH|EGSC|EGSS|EGTE|EGTO|EGVP) ||**

**(!(ARR ~ /EBAW|EBBR|EBCI|EBCV|EBMB|EBLG|LFPB|LFPC|LFPG|LFPN|LFPO|LFPT|LFPV|EH../))) {REGLE = REGLE " - DET VABIK"}**

### Alternativ Compulsory Point PT1/PT2:

**if ( ! (VIA ~ /PT1/)) { REGLE = REGLE ">PT2" }**

**if ( ! (VIA ~ /PT2/)) { REGLE = REGLE ">PT1" }**

## RAD Type Restriction

There are three main types of restrictions, the inclusive, exclusive and compulsory, examples of which are given below:

**INCLUSIVE** restriction

UN869 LERGA-OLRAK

Not available for traffic

Dep. LIML

With Dest. LICJ

Traffic must meet ALL of the conditions to be subject to the restriction

SAAM

```
if (DEP ~ /LIML/) && (ARR ~ /LICJ/) { REGLE = REGLE " - BOL TAQ" }
```

**EXCLUSIVE** restriction

UL976 OBATO - RONAX

Not available for traffic

1. Dep. LIML

2. Dest. LICJ

3. Via ARNOS

Traffic only needs to meet ONE of the conditions to be subject to the restriction.

SAAM

```
if (DEP ~ /LIML/) || (ARR ~ /LICJ/) || (VIA ~ /ARNOS/) { REGLE = REGLE " - OBATO RONAX" }
```

or

```
if (DEP ~ /LIML/) { REGLE = REGLE " - OBATO RONAX" }
```

```
if (ARR ~ /LICJ/) { REGLE = REGLE " - OBATO RONAX" }
```

```
if (VIA ~ /ARNOS/) { REGLE = REGLE " - OBATO RONAX" }
```

**COMPULSORY** restriction

UL865 BOL - TAQ

Compulsory for traffic

Dep. LIML

With Dest. LICJ

Traffic has no other option; it shall fly this route segment

SAAM

```
if (DEP ~ /LIML/) && (ARR ~ /LICJ/) { REGLE = REGLE " > BOL TAQ" }
```

## 4.7 Assignment data validation process

### Introduction

Standard [assignment](#) output files are a \*.zin file containing 2D trajectory of assigned flights and \*\_missing.exp file containing the list of flights that could not be assigned.

Even if missing flights can sometimes be reinjected from original traffic data as explained [here](#), it is important to keep the number of missing flights as low as possible, and anyway, to precisely know why some flights are missing.

At a smaller extent, it is useful to have information on the constraints (RAD, turn angles, profile restrictions, ...) that may have been met during the assignment process in order to assess the impact of such constraints on the traffic flows.

Analyzing and managing this information is the assignment data validation process and enables a deeper understanding of your airspace design.

### Process

Below are listed the different steps for a simple standard validation process :

#### 1. Cleaning of assignment ERRORS/WARNINGS

Before even analyzing the assignment outputs, it is needed to clean or understand the error and warning messages produced during the assignment. They appear in the assignment .log file.

#### 2. Running the assignment with no additional outputs

A \*.zin or \*\_2d.so6 file is produced for assigned flights and \*\_missing.exp for missing flights. You may check that the number of assigned flights plus the number of missing flights equals the number of original flights in your traffic demand file.

#### 3. Running the assignment on \*\_missing.exp file with additional outputs

In addition to the standard assignment output files, this will produce a bunch of additional files containing information helping to understand why the flights could not be assigned.



## Additional output files

### 1. Splitting of missing flights

A flight can be missing for two reasons mainly : either it is or would be too long regarding allowed extension, or no trajectory can be found at all.

These "too long" flights are output in a 2D so6 file. The trajectories that "would be" too long actually are trajectories that already are too long, but that do not satisfy all the constraints. The assignment process stopped the resolution to save resources. More precisely, the assignment stops when the route length extension reaches twice the allowed extension. These are the "Nok" trajectories. These "Nok" trajectories may reveal problems or conflicts with the different constraints. Have a look at the related constraints files !

In addition, each of this "too long" or "nok" trajectories are split in 2 separate files depending on whether the flight is circular (ADEP=ADES) or not. All these 2d trajectories can be displayed in the [Queries](#).

When no flight path can be found at all, the missing flight is listed into a \*\_missing.exp file which is an extract from the expand file given in input of the assignment process. Two different missing files are used :

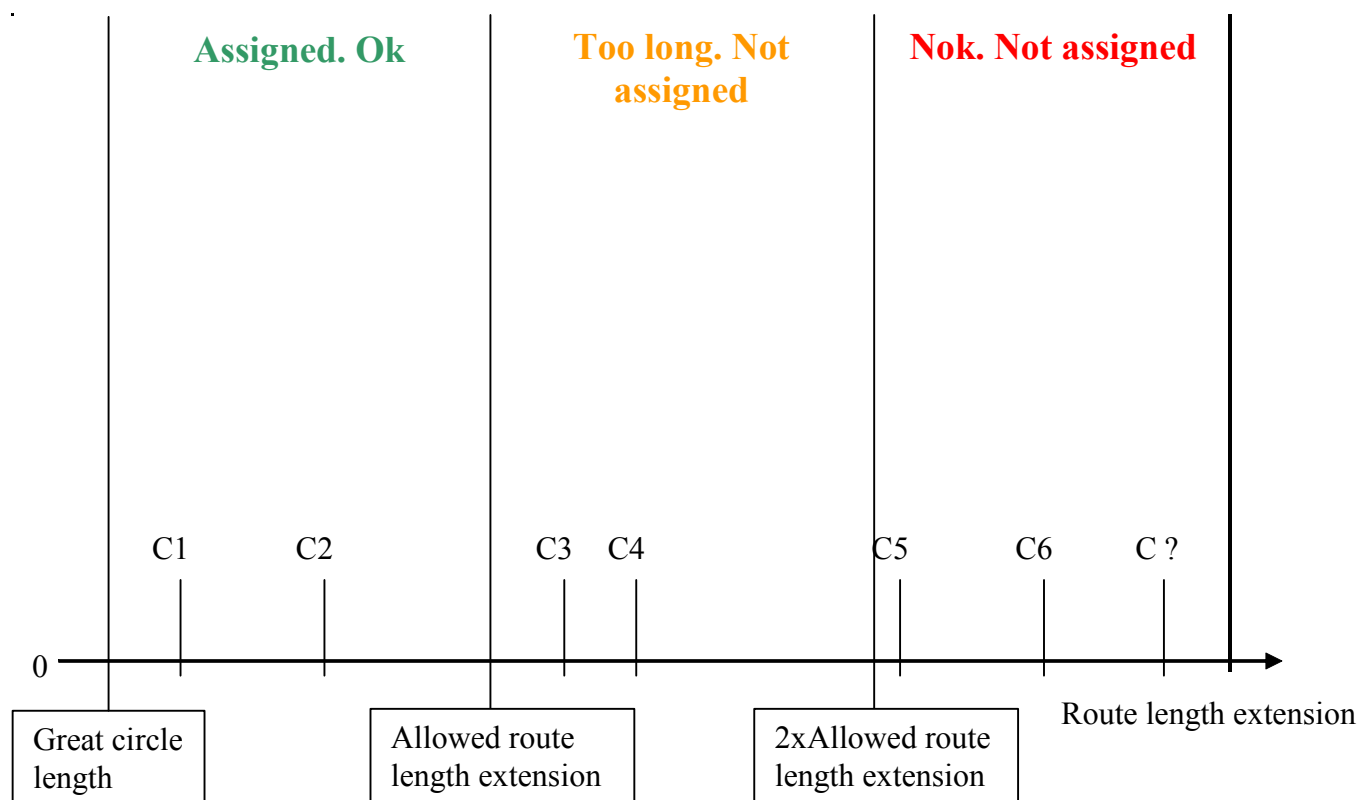
- AFIL\_ZZZZ\_missing file that lists all flights having either AFIL or ZZZZ as a departure or arrival airport for which no coordinates are available, hence, no route can be defined,
- connectivity\_missing file that lists all flights for which not even a single path from origin to destination can be found across the network, no matter the constraints are. This indicates a poor network connectivity, or a poor network definition around an airport.

At last, you may be interested in checking the number of flights : the sum of the number of flights in the different additional 2D so6 output files and in the different specific missing expand files should equal the total number of missing flights given by the assignment.

### 2. Constraints files

Constraints files are text files which contain one flight constraints information per line. A line lists some constraints C1, C2, C3, ... related to the flight identified in the beginning of the line (origin, destination and flight ID).

- ...\_assigned.txt file that tell which constraints have been met and solved during the assignment process, resulting in a higher route length extension. Note that the constraints that were naturally satisfied (i.e. without any extra route length extension) are not listed.
- ...\_not\_assigned.txt files that tell a set of constraints that had to be solved. Trying to solve them put the route length extension above the allowed route length extension for the too long trajectories and above twice the allowed route length extension for the nok trajectories. Therefore, nok trajectories may still have additional unsolved constraints not listed in this constraint file. For nok trajectories, the last listed constraint is unsolved in the associated 2D so6 file and can be visualized thanks to the [Queries](#).



### 3. Summary of the additional output files

		Trajectories files		Constraints files
		2D.so6	missing.exp	
Flights assigned OK		normal output	N/A	present
Missing flights	Too long trajectories	not circular	N/A	present
		circular		present
	Nok trajectories	not circular		present
		circular		present
	No trajectory	N/A	AFIL/ZZZZ connectivity problem	N/A
Total number of flights	Total number of flights	Total number of flights		

## Constraints description

Here is the explanation of the different constraints information that can be found in

constraints files:

#### **+/-\_MAVAR\_ALELU**

This means that segment MAVAR\_ALELU was preferably chosen but is forbidden for this traffic flow or reserved for another traffic flow.

#### **+/-\_MAVAR**

This means that point MAVAR was preferably chosen but is forbidden for this traffic flow or reserved for another traffic flow.

#### **\*\_HOLLY\\_/\_HOLLY\_WILLO\\_/\_WILLO\_\***

This means that a turn to segment HOLLY\_WILLO or from segment HOLLY\_WILLO was preferably chosen but the turn angle is forbidden.

#### **LEVEL/TIME\_BEGAR\_ODINA**

This means that segment BEGAR\_ODINA was preferably chosen for routing but flight profile is not compliant with allowed level band or opening time for that segment. (see [profile restrictions during assignment](#))

#### **!RISLA\_KEPAD,KEPAD\_HON|!LIFFY**

This means either that segment RISLA\_KEPAD or KEPAD\_HON was preferably chosen but in a route with point LIFFY overflown whereas an existing [segment management rule](#) specify that for this traffic flow, going both via point LIFFY and segment RISLA\_KEPAD or KEPAD\_HON is forbidden. The solution to this is either to go only via LIFFY, or RISLA\_KEPAD/KEPAD\_HON without overflying LIFFY, or not going via LIFFY nor RISLA\_KEPAD/KEPAD\_HON, depending on what is the shortest solution.

#### **>KOKOS,GODIX,ROLEN|!EVX\_LGL**

This means either that point KOKOS, GODIX or ROLEN was not preferably chosen but in a route with segment EVX\_LGL overflown whereas an existing [segment management rule](#) specify that for this traffic flow, going via segment EVX\_LGL implies that the flight overflies either KOKOS, GODIX or ROLEN as well. The solution to this is either not to go via KOKOS, GODIX or ROLEN, or (inclusively) not to go via EVX\_LGL, or to go both via KOKOS, GODIX, or ROLEN, and EVX\_LGL, depending on what is the shortest solution.

#### **>KUDOP**

This means that point KUDOP was not preferably chosen but is mandatory.

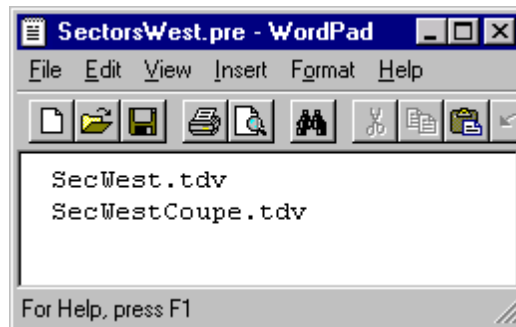
## **Remarks**

- if an atlantic flight is listed in the "Nok" trajectories and if a lot of LEVEL/TIME constraints appear in the penalizing\_and\_solved\_constraints file, check your North atlantic Free route network, it may be that the flight RFL is above the upper limit of the Free route network.
- it is of course possible to get additional output validation files for flights that are correctly assigned and check the constraints that have been met and solved.

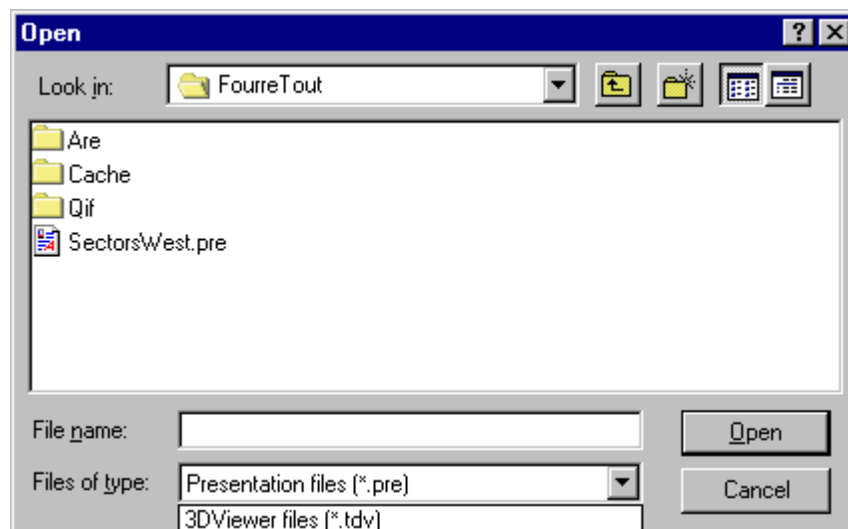
- it may turn out that some RAD restrictions systematically are "naturally" satisfied and need not to exist as such in the RAD.

## 4.8 Preparing a SAAM Presentation

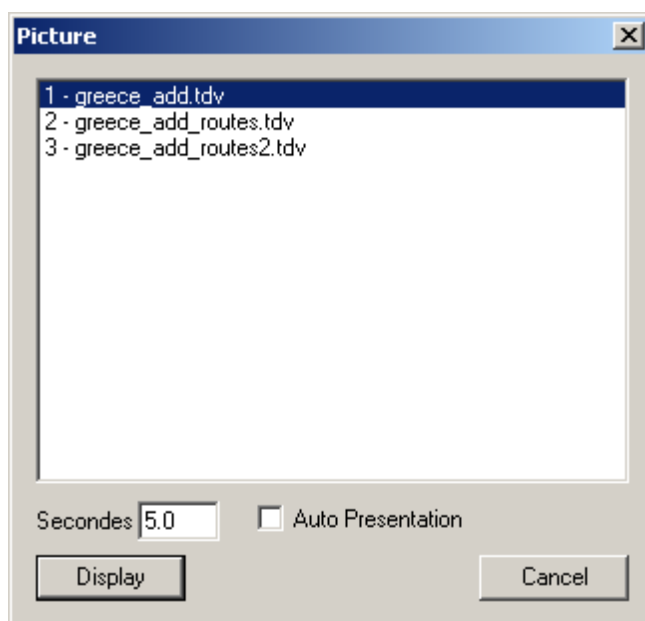
A presentation is a set of SAAM tdv's. They are stored in a text file. To create a presentation file (\*.pre), use a text editor such as notepad.



Once the text file is completed, load the presentation using the standard 'open' feature but choose \*.pre instead of \*.tdv.



To switch from one Tdv to the other use the **Page Up/Page Down** buttons.



To have an overview of the presentation use the presentation icon 

To switch the Tdv's automatically, use the '**Auto Presentation**' and define a time-frame in seconds.

## 4.9 Creating a SAAM animation

### Overview

SAAM provides a mechanism to create complex animations of SAAM objects and traffic:

- The animation of objects (sector, title, images...) makes the objects appear and disappear from the SAAM viewport.
- The animation of traffic shows the airplanes moving along their trajectory.

An animation is defined in your TDV file. It uses the following commands:

- a **CLOCKx** command that defines the time the animation starts and ends and the speed of the animation
- one or more **SANIM** commands to animate airplanes from a traffic file
- one or more **CURVE** commands to animate SAAM objects such as sectors, text titles, images ...

These commands can be typed manually in the [TDV with a text editor](#) or, for the **CLOCKx** and **SANIM** commands, generated automatically with dialog boxes. We will concentrate here on the syntax of the **CURVE** commands. For the other commands, use preferably the dialog boxes: [Add Traffic animation](#).

However you created the animation commands, SAAM will animate all objects and the traffic simultaneously, following the speed of the **CLOCKx**.

If you want to disable the animation of a specific object or of a traffic, simply put a **#** in front of the command in the TDV. To disable the animation altogether, put a comment in front of the **CLOCKx** command.

Before an object can be animated, it must be present in the TDV file. Please refer to: [Adding graphical objects](#), [Add Airspace](#), [Add Network](#), [Add Circle](#), [Add Traffic Animation](#)

See also: [Using the TDV Editor](#)

## Timeline

To define the evolution of an object during the time of the animation, a timeline for each object is required. A timeline defines a list of (simple) actions that take place at a given time. For a traffic animation, the timeline is defined in the traffic file itself: the position of the airplane along the time. For non moving objects, the timeline is defined in a separate file (extension .tim). A tim file specifies the visibility of the object with time and optionally the evolution of other properties e.g. the modification of a sector color.

**Warning:** there is no Graphical User Interface (GUI) to define an non-traffic animation (**CURVE** command): you have to add manually the **CURVE** command in your TDV file and create a time line (tim file) with a text editor.

## Basic steps to create an animation

Typically you will do the following to create an animation:

1. Add all the objects that must be animated in your study (e.g. [Adding graphical objects](#), [Add Airspace](#), [Add Network](#) ...)
2. Add one or more Traffic animation via the Add Traffic Animation dialog boxes. This will add the **CLOCKx** and the **SANIM** commands in your TDV.
3. Define the timeline of your non traffic objects: use a text editor to define when your objects are visible ... Save this information in a .tim file
4. Add the animations of the non-moving objects: edit your TDV (but before don't forget to save it!) and add one or several **CURVE** commands. Save and close your TDV. The animation will start
5. To stop the animation of an item, edit the TDV and place a # in front of its **CURVE** command. Close your TDV. To stop the animation altogether, edit the TDV and place a # in front of the **CLOCKx** command

The syntax of the animation commands and of the timeline commands are defined below.

## My first animation

Let's look how to create a simple animation to show a sector from 12.00 until 14.00 (without a traffic animation).

1. The object to animate must be present in your TDV file: add a sector ([Add-> Airspace](#)) and save your tdv. (in our example we are animating the sector EDMMN)
2. Create a tim file (with a text editor) that will indicate the sequence of appearance of your object. Call it mysectors.tim. Save the file in the same directory as your tdv's directory. The file contains the list of actions on the object: a time (in seconds) in the first column and the action in the second: 0 means hide, 1 means display.

```
Time Sector
0 0
43200 1
50100 0
```

3. Edit the TDV to add an animation. Add one line to indicate you want an animation (**CLOCK**) and one line to specify which object you want animate (**CURVE**)

```
CLOCKx 1 43200.00 50400.00 20.000000 LOOP
```

```
CURVE anim mysectors.tim 0 1 1 1.000000 N EDMN.are C 64 255 0 0
```

4. Save your TDV file and close the text editor.
5. SAAM will start the animation: your sector will be visible from 12.00 until 14.00

### Objects that can be animated

You can animate the following SAAM (non-moving) objects: line, point, title, network, image, text, 3D volumes (airblocks, prism,..) and the camera and the zoom. The type of animation depends on the type of object. For a network, the action is either show the network or hide it. For a sector you can specify how its color changes with time. The table below specified for each object, the type of animation supported.

**Reminder:** the objects to be animated must be present in SAAM viewport before they can be animated. See [Adding graphical objects](#), [Add Airspace](#), [Add Network](#), [Add Circle](#)

Objects that can be animated			
TDV object	object_name	effect is defined by the n value = (value read from the tim file) * (factor)	comment
L (line)	segment_name	0: disable, <>0: enable. The thickness of the line varies with the value of n	Max 800 different L objects having the same segment_name can be linked into a single CURVE command
P (point)	point_name	0: disable, <>0: enable. The (thickness + size + height) varies with n	Max 800 different P objects having the same point_name can be linked into a single CURVE
TITLE	text_to_display	0: disable, <>0: enable	n value can be displayed in the title if '#' tag if found in "text_to_display" (idem for '^' tag for time display)
NETWORK	ase_file	0: disable, <>0: enable	
IMAGE	bmp_file	0: disable, <>0 : enable. The size of image varies with n	n=100 gives size given in IMAGE TDV command (if, in the IMAGE TDV command the size_factor is 1.0 then, with n=100, the size is the original)
D or Dx (designator)	text_to_display	0: disable, <>0: enable	
V or Vx (volume)	file_name.are	0: disable, <>0: enable. The color of the volume varies with n	linear interpolation between TRGB color components of the Volume and the color defined in CURVE. n=0 represents the Color defined in the CURVE n=100 represents the Volume color.
CURVE	title_of_curve	depends on object to which CURVE is linked (see above and below this row)	Several CURVE objects (number depends on memory size) having the same title_of_curve name can be linked into a single CURVE command. The value found in the 2 TIM files will be multiplied, and the result will be applied to animated objects.
Y MOVE (camera)	Y MOVE	n defines the XMOVE of the camera	goes in south/north direction (movement absorber is included)
X MOVE (camera)	X MOVE	n defines the YMOVE of the camera	goes in west/east direction (movement absorber is included)
Y ORIENT (camera)	Y ORIENT	n defines the XORIENT of the camera	change tilt angle (movement absorber is included)
X ORIENT (camera)	X ORIENT	n defines the XORIENT of the camera	change azimuth angle (movement absorber is included)
ZOOM	ZOOM	n defines the ZOOM of the camera	change zoom (movement absorber is

Objects that can be animated			
(camera)			included)

### Defining the time line (tim file)

A timeline is defined for each object: it indicates when an object is displayed. In addition it can also specify how the object properties change in time (eg its color). This information is stored in a text file with the extension tim.

Example:

```
Time object1 object2
0 0 0
25200 1 0
25400 100 1
25600 1 1
25800 100 1
26000 1 0
26200 100 1
```

The first field indicates the time (can be defined in seconds, HHMM ...), the second field the action. 0 means *Don't display*. Anything else means display the object. For possible "display" values, refer to the table above.

You can share the timeline of several object in a single file: each column after the time column represents the timeline of an object.

In addition, you can specify that an action is only executed at a given clock turn. (the clock turn is the number of times the animation has run). To achieve this, your tim file should contain at column 2, a column named **NB\_CLOCK\_TURN**. Note: this particular column does not affect tim\_file\_column numbering e.g. the time\_file\_column for object1 will stay 1.

Example: tim file with NB\_CLOCK\_TURN, time express in HHMMSS using tab separator (represented as →)

```
time→NB_CLOCK_TURN→myzoom→cam_YMOVE
000000→2→1246149→-27759
013000→2→1246149→-27759
020000→2→1246149→-27759
060000→2→300000→2258
140000→2→300000→2258
235959→2→50000→2258
```

The NB\_CLOCK\_TURN 2 means that the action will only be done at the second clock turn. Outside turn 2, the action will not be seen by SAAM. Note: the last action of the active clock turn will define how the objects will be seen in the following clock turns. Example: you add an action to show gradually a sector starting at 15.00 and being fully visible (Solid color) at 16.00; the sector will stay fully visible for the rest of the animation (turn clock 3 and after).

See also: [tim - Timeline file format](#).

### Defining the CURVE animation in your TDV

Each object to animate requires a one line command (**CURVE** command) in your TDV file.



In addition to the display of an object, the **CURVE** command can also plots a graph that shows the evolution of the *n* value with time.

The syntax of the command is:

```
CURVE title_of_curve timeline_file.tim graph_position graph_text_position tim_file_obj
factor graph_display_flag object COLOR
```

- **title\_of\_curve:** it is the name of the animation. It is used: 1) as the label of the optional graph. 2) to visually distinguish CURVES in the TDV. 3) to be linked to another CURVE object, it is possible to provide the same title for several curve, in such case, all these CURVE might be linked to another CURVE. In the example below, the CURVE "move" is animating 4 CURVE "SECT":

```
CURVE move Sectors.tim 0 1 4 1.000000 N SECT C 255 255 128 0

CURVE SECT ANIM_SELECT/EBBUEEC_150410.tim 0 1 1H 1.0 N MY_SELECT_collapse.gar:EBBUEEC(
CURVE SECT ANIM_SELECT/EBBUEHS_150410.tim 0 1 1H 1.0 N MY_SELECT.gar:EBBUEHS C 64 85 1
CURVE SECT ANIM_SELECT/EBBUELS_150410.tim 0 1 1H 1.0 N MY_SELECT.gar:EBBUELS C 64 217
CURVE SECT ANIM_SELECT/EBBUESC_150410.tim 0 1 1H 1.0 N MY_SELECT_collapse.gar:EBBUESC(
```

- **timeline\_file.tim:** the file defining the display status of the object with time
- **graph\_position:** the position of the graph: 0 (left) 1 (left-center) 2 (center) 3 (right). In the example above the graph is not displayed (N for 8th parameter)
- **graph\_text\_position:** the position of the X axis label of the graph: 1 to 6
- **tim\_file\_object\_index:** column number where data must be read in the TIM file, ranges from 1 (=first column after the time column or NB\_CLOCK\_TURN column) to max 100
- **factor:** allows to expand or reduce the value read from TIM file. Factor = 1 means max value found in selected column of the tim file (so reaching top of diagram) is 100.
- **graph\_display\_flag:** show the graph or not. Values: Y or N
- **object:** either "NULL" (no object attached, only a graph is displayed) or the name of an object present in the TDV file (see column 2 "object\_name" in the table above)
- **COLOR:** the color that corresponds to *n*=0. Syntax: "C Transparent Red Green Blue"  
Each TRGB integer value can vary from 0 to 255.  
Transparent=0 means invisible. Transparent=255 means solid color.

## Recording an animation

Different ways are possible.

- Tool to capture in real time content of SAAM screen animation and storing the result in an AVI file is one possibility (see for example free Hypercam software on internet). The disadvantage is the poor resolution and low frame rate and even sometimes bad quality of the picture due to too high compression ratio, and it is really dependant on your computer speed and number of processors (at least 2 is mandatory). But it is easy, quick and convenient for small movies to be inserted on a PowerPoint presentation for instance.
- SAAM provide another mean for movie recording by generating and storing a numbered png picture file on disk for each generated frame of your SAAM animation. If the animation is long lasting, thousands of pictures could be generated in your working directory. The size of SAAM screen can be bigger than your actual screen (try menu View/Set screen size to video size), it works well allowing to generates images having a higher resolution than your

flat screen allows it (but depends also your your graphics card memory capacity) ! No frame will be missed, revolution is kept and then the resulting quality is excellent. The inconvenient is to use a separate tool to convert all generated images into a proper movie, but such tool exists. The shortcut command to trigger the generation of pictures for each frame is: ALT-R (like Record) to start it (it asks a confirmation) and also ALT-R to stop it (it indicates the number of images that were generated).

## 4.10 Adding graphical objects

You can add non-ATM graphical objects (shapes, images (jpg), title ....) to the SAAM viewport to enhance its look.

To add a graphics object, you need to [edit the TDV file](#) manually with a text editor and add a command to specify the type of object and its properties: color, size .... There is no GUI to achieve this, except for some objects (see below).

To animate the graphics objects, please refer to [Creating a SAAM animation](#)

To add ATM objects, please view [Add Menu](#)

### Line

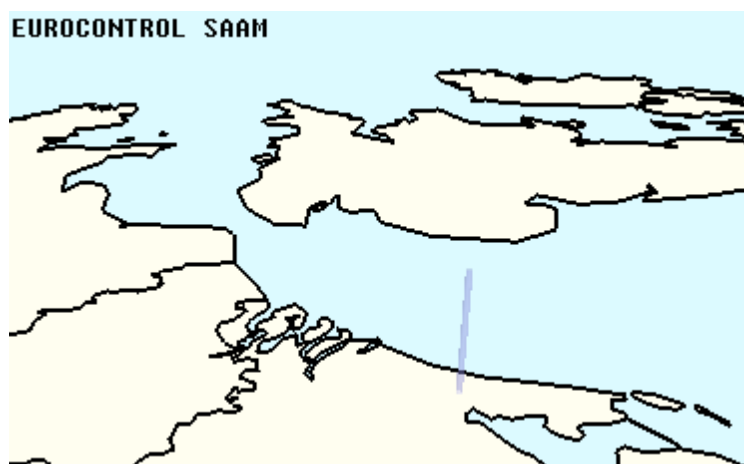
A SAAM line is a 3D line (prism) with a predefined contour (point, triangle,...). The line can be vertical or tilted.

TDV Syntax:

```
L S_T line_name line_shape lat1 long1 alt1 lat2 long2 alt2 thickness COLOR
```

Example:

```
L T My_Line 3 3140.433000 285.483300 70.000000 3138.983000 286.350000 0.000000 90.0000
```



A line

Typical usage:

#### ☐ The line parameters

**S\_T:** **S** for a solid line, **T** for a transparent line. The level of transparency is defined by the **COLOR** parameter (see below)

**line\_name:** the name is used in animation commands. It is not displayable on the

SAAM viewport.

**line\_shape:** defines the section of the line and the presence of an arrow. Possible values:

Line Section	Line with no arrow	Line with an arrow in the middle	Line with an arrow at the end
line (one pixel)	0	20	40
triangle section	1	21	41
square section	2	22	42
circle section (low def: polygon with 8 vertices)	3	23	43
circle section (med. def.: polygon with 16 vertices)	4	24	44
circle section (high def.: polygon with 32 vertices)	5	25	45

**lat1 long1 lat2 long2:** coordinates in minute decimal of the end points of the line

**alt1 alt2:** Decimal Flight Level of the end points of the line

**COLOR:** = "C Transparent Red Green Blue"

Each TRGB integer value can vary from 0 to 255.

Transparent=0 means invisible. Transparent=255 means solid color. If the S\_T parameter is S(olid), the transparent value has no effect.

## Point

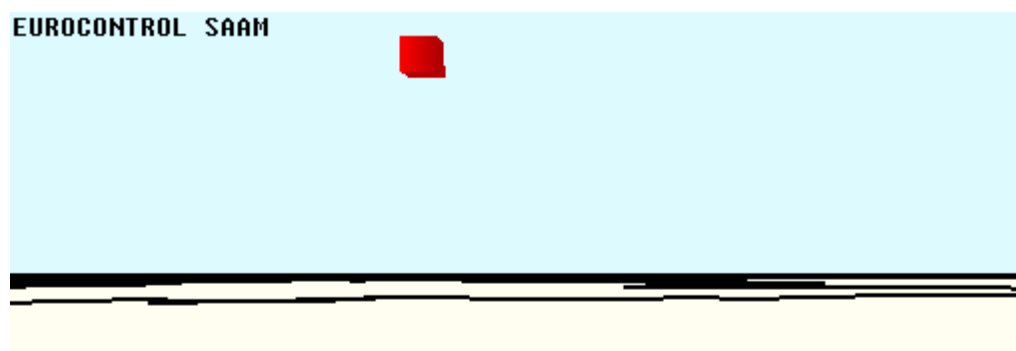
A SAAM point is a 3D point (vertical prism, vertical cone or vertical starfish) that can have a height and thickness.

TDV Syntax:

```
P S_T point_label point_shape lat lon alt_base height thickness COLOR
```

Example:

```
P S EBBR 2 3054.633000 269.666700 5.000000 .500000 20.000000 C 64 255 0 0
```



A point (a red square cylinder)

Note: A point of type prism is similar to a SAAM Line except that a prism point is always vertical.

### ▣ The point parameters:

**S\_T:** **S** for a solid point, **T** for a transparent point. The level of transparency is defined by the **COLOR** parameter (see below)

**name:** the name is used in animation commands. It is not displayable on the SAAM viewport.

#### **point\_shape:**

- 0= point section (one pixel),
- 1= triangle section,
- 2= square section,
- 3= circle section (low definition: polygon with 8 vertices),
- 4= circle section (medium def.: polygon with 16 vertices),
- 5= circle section (high def. : polygon with 32 vertices)
- 6= up cone (low def.),
- 7= up cone (medium def.),
- 8= up cone (high def.)
- 9=Up Starfish 3 branches,
- 10=Up Starfish 4 branches,
- 11=Up Starfish 5 branches,
- 12=Up Starfish 6 branches,
- 13=Up Starfish 8 branches

**lat lon :** coordinates of the point in minute decimal

**alt\_base:** vertical position of the point in Decimal Flight Level

**height:** height of the point/cylinder in nautical miles decimal

**thickness:** horizontal thickness of the point/cylinder. By default in internal unit, except when "nm" is added at the end of the thickness in that case it is expressed in nautical mile

**COLOR:** = "C Transparent Red Green Blue"

Each TRGB integer value can vary from 0 to 255.

Transparent=0 means invisible. Transparent=255 means solid color. If the **S\_T** parameter is **S**(olid), the transparent value has no effect.

## Vertical prism

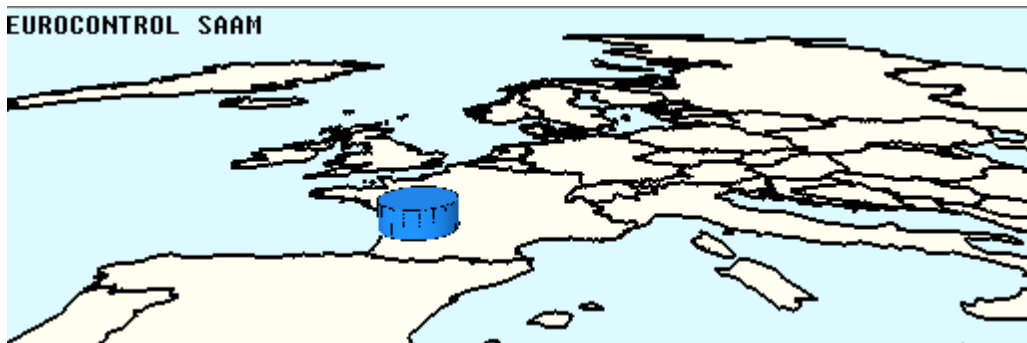
You can add a vertical prism whose contour/section is defined by an [are file](#). Although you can add a prism with any type of contour, the most typical usage is to add a cylinder with a circle section. To achieve this, SAAM provides a dialog box: see [Add Circle](#). Tip: if you want to use another section, create a circle cylinder with the dialog box and edit its shape with the [airspace volume editor](#) or replace, in the TDV, the circle.are file by the are file defining your specific contour.

Syntax:

```
| V S_T L_N_T_OB_X_CI_X contour.are bot_level top_level COLOR(*)
```

Example:

```
V T LXI EHRD_circle.are 999.00 999.00 C 0 0 0 100
```



A vertical cylinder created with a Add circle dialog box

**Note:** [airblocks](#) are displayed with the same TDV prism command.

### Parameters:

Since the the TDV command for the cylinder can be created and changed with the Add Circle dialog box, the parameters of the TDV command is not explained here.

## Vertical quadrilateral prism

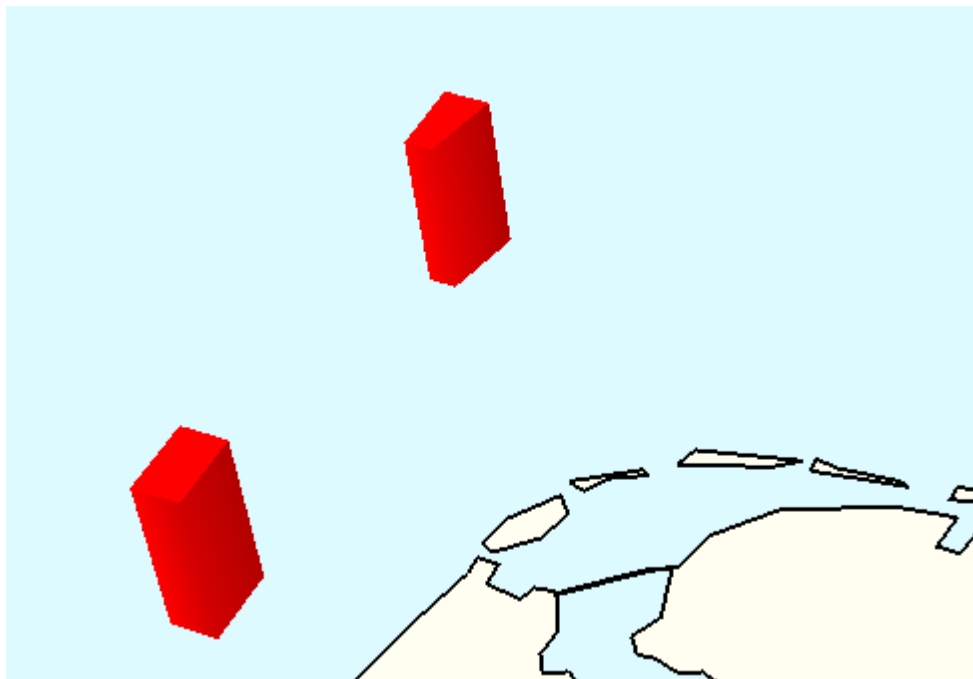
You can add to the viewport a vertical quadrilateral prism. Typically you will add either a cube or a parallelepiped but SAAM accepts any kind of quadrilateral shape. This shape is a sub-class of the vertical prism (advantage: you don't need to define an are file).

Syntax:

```
C S_T name lat1 lon1 lat2 lon2 lat3 lon3 lat4 lon4 bot_level top_level COLOR
```

Example:

```
C S my_Parallelepiped 3140.0 258.0 3150.0 258.0 3150.0 268.0 3140.0 268.0 50 100 C 64
C S my_quadrilateral_prism 3200.0 258.0 3210.0 258.0 3210.0 268.0 3200.0 260.0 50 100
```



A Parallelepiped and quadrilateral prism

#### Parameters of the vertical quadrilateral prism

**S\_T:** **S** for a solid point, **T** for a transparent point. The level of transparency is defined by the **COLOR** parameter (see below)

**name:** the name is used in animation commands. It is not displayable on the SAAM viewport.

**lat1 lon1 lat2 lon2 lat3 lon3 lat4 lon4:** coordinates of the 4 vertices of the base quadrilateral (in minute decimal)

**bot\_level top\_level:** bottom level and top level of the prism (in Decimal Flight Level)

**COLOR:** = "C Transparent Red Green Blue"

Each TRGB integer value can vary from 0 to 255.

Transparent=0 means invisible. Transparent=255 means solid color. If the **S\_T** parameter is **S**(olid), the transparent value has no effect.

## Labels (designator)

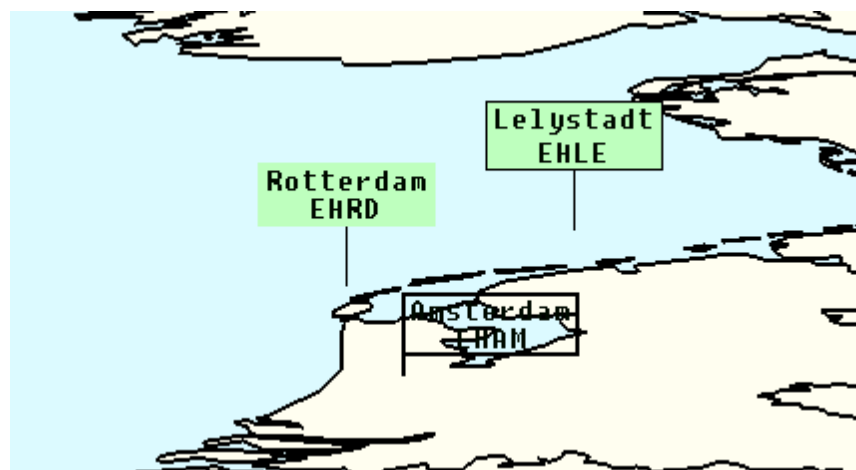
To display text (one or two lines) anywhere in your airspace. You can specify the color of the background, the alignment of the text, add a post ....

Syntax:

```
Dx version(=1) text@to@display|2nd@line lat lon alt feature_version(=1) aspect(=F/
frame_thick(=0/1) text_alignment(=L/C/R) post(=0/1) post_shiftX post_shiftY
post_alignment(L/C/R) font_logical_name(*) auto_color_text(0/1) COLOR_text(*)
auto_color_frame(0/1) COLOR_frame(*) auto_color_backgrd(0/1) COLOR_background(*)
```

Examples:

```
Dx 1 Amsterdam|EHAM 3138.430000 285.260000 0.000000 1 F 1 C 1 0 10 L label 0 C 64 0 10
Dx 1 Rotterdam|EHRD 3117.430000 266.130000 40.000000 1 B 0 C 1 0 30 C label 0 C 64 0 1
Dx 1 Lelystadt|EHLE 3147.160000 331.660000 40.000000 1 T 0 C 1 0 30 C label 0 C 64 0 1
```



### ▣ Label parameters

**version:** the version of this command. Currently only one version = 1

**text@to@display**

**text@to@display|2nd@line:** text to display. One or two lines separated by |

**lat lon alt:** coordinates of the bottom point of the post. (lat , long in minutes decimal, alt in FL decimal)

**feature\_version:** always 1

**aspect:** F=Frame, B=Background, T=Two(frame+background), X=(no frame and no background)

**frame\_thick** tickness of the frame: 0=thin 1=thick

**text\_alignment** =L=left C=center R=right

**post** 1=display a post under the text box: line below the bottom line of the text box.  
0=no post

**post\_shiftX post\_shiftY** shift from the bottom coordinate of the post. If post\_shiftX is different to 0, the post will be tilted. Express in decimal FL.

**post\_alignment** the post is connected to the text box on the L=left C=center R=right of the lower side of the box

**font\_logical\_name** is the logical name given in the TDV FONT command else it can be "DEFAULT" for a Fixedsys500c.ttf font of size 12 normal thickness (which is installed automatically by SAAM)

**auto\_color\_text** (0/1)

**COLOR\_text** = "C Transparent Red Green Blue"

Each TRGB integer value can vary from 0 to 255. Transparent=0 means invisible. Transparent=255 means solid color.

**auto\_color\_frame** (0/1)

**COLOR\_frame** see COLOR\_text

**auto\_color\_backgrd** (0/1)

**COLOR\_background** see COLOR\_text

## Title

Display a line text anywhere on the viewport. Contrary to the labels that are position in 3D in the airspace, the titles have a static position on the viewport: they do not move when you change your camera.

You can add as many titles as you want: at the top, middle, at the bottom of the screen.

Syntax:

```
version=1 aspect(S/T) text_to_display size_of_the_font screen_coord_x screen_coord_y
version=2 aspect(S/T) text_to_display size_of_the_font screen_coord_x screen_coord_y
```

Example:

```
TITLE 2 S NAVIGATION 30 10 850 0 0 C 64 255 128 0
TITLE 2 S Airport/TMA@Improvement:@Amsterdam@Area 50.000000 150 50 2 2 C 64 255 128 0
TITLE 2 S Towards@one@TMA 50.000000 400 800 2 2 C 64 255 128 0
```

### The Title parameters

**version:** version of the command: 1 or 2

**aspect:** **S** for a solid text, **T** for a transparent text. The level of transparency is defined by the **COLOR** parameter (see below)

**text\_to\_display:** text to display. Respect the following conventions:

- use **@** instead of blanks e.g. DYNAMIC@REROUTING
- **#** =numbering part\*coef coming from CURVE for animation,
- **^** =current clock time does not more require CURVE

**size\_of\_the\_font:**

**screen\_coord\_x, screen\_coord\_y:**

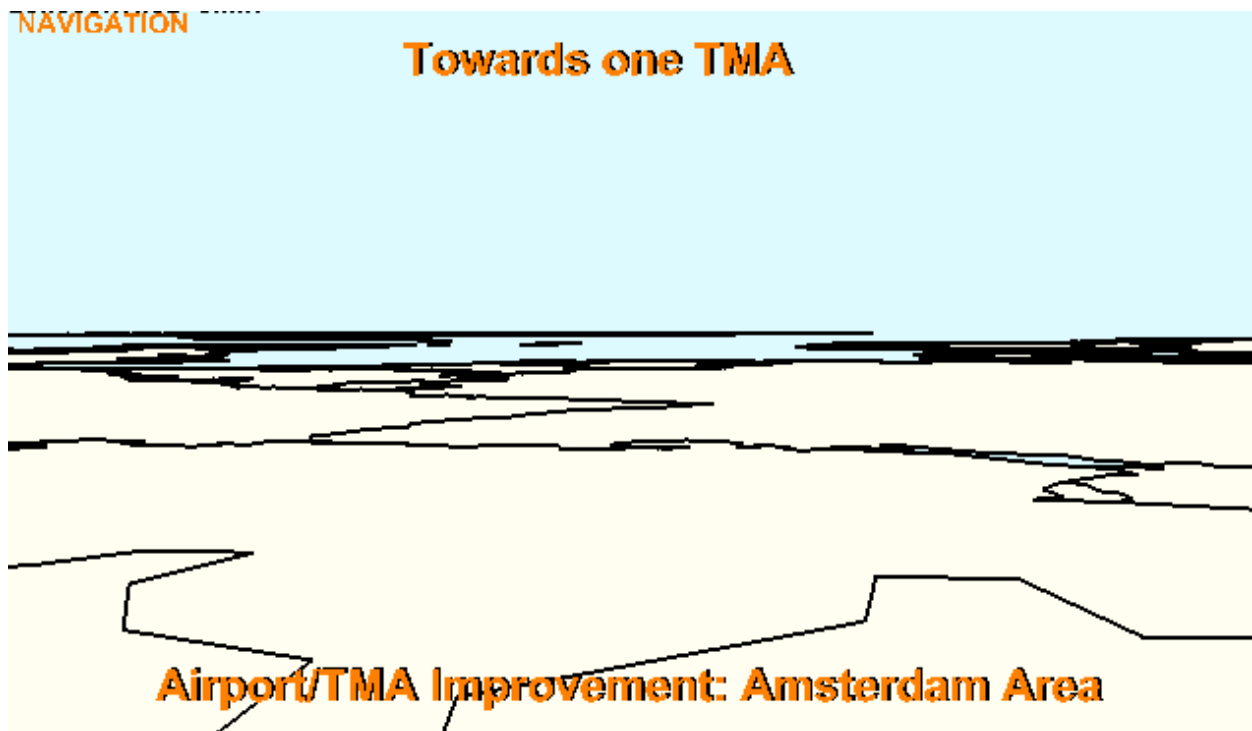
**shadow\_x, shadow\_y:**

**COLOR:** = "C Transparent Red Green Blue"

Each TRGB integer value can vary from 0 to 255.

Transparent=0 means invisible. Transparent=255 means solid color. If the S\_T parameter is **S**(olid), the transparent value has no effect.





## Image

Add an image to your viewport. It can be treated as a flag so that it moves in 3D according to the position of the SAAM camera (in a similar fashion as a Flag) or it can stick to a fix position on SAAM viewport (similar to a Title).

Syntax:

```
Image version=3 aspect=S 3D_SCENE(default)|SCREEN file_name.bmp size_factor origin lat
```

Example:

```
IMAGE 3 S 3D_SCENE BlueMedLogo.bmp 1.00 LB 2552.00 767.00 600.00
IMAGE 3 S SCREEN HCAALogo.bmp 2.00 LB 100 100 600.00
```



## Parameters

**version=3**

**aspect= S** means Solid (Transparent is not available)

**3D\_SCENE(default)|SCREEN.** If 3D\_SCENE then the bitmap is displayed in the scene like a flag in the air, else it is displayed at a fixed position on the SCREEN (Saam viewport): its position does not change with the SAAM camera position. Note: in 3D\_SCENE, the image disappears if any part of it becomes invisible.

**file\_name.bmp** the name of the file that contains the image. Supported format: bmp only.

**size\_factor:** in decimal; 0=very very small, 1.0=original image size, 2.0 twice as big .... Warning: scaling will degrade the image precision seriously.

**origin:** specifies the anchor point of the image for the the lat, long, alt. 2 letters: x-axis={Left, Centre, Right}, y-axis={Top, Center, Bottom}, example: CB means that lat, long, alt will refer to the center bottom point of the image.

**lat lon alt:** If 3D\_SCENE: lat, long, alt specify the position of the **origin** of the image ( bitmap origin is centered in the lower part)

If SCREEN: lat & lon specify the location of the **origin** of the image in x and y coordinate of the screen (screen pixels). x=0 and y=0 is position at the lower left corner of viewport. alt is not used.

## 4.11 Airspace Activation Editor

### Accessed by

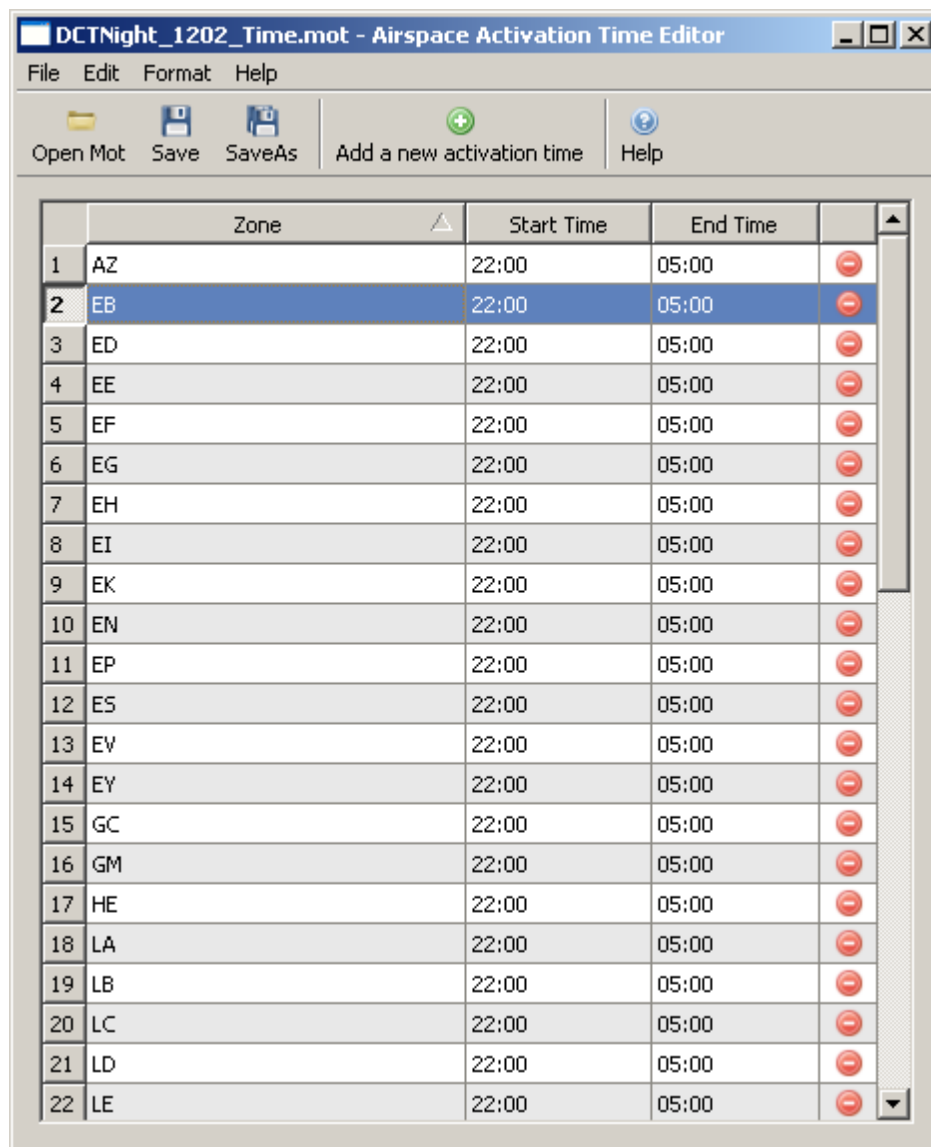
The editor can be launched from the [assignment](#) module and from the [Route Choice Based on Military Opening Time](#) dialog.

### Purpose/description

View and edit Airspace Activation Times files (\*.mot). It provides information on airspace activation, military activities. If a military zone is not present in the table it is assumed to be not active and open to civil traffic for the [Route Choice Based On Military](#) process.

The editor need to load data from a sector file (\*.sls or \*.gsl) but this file cannot be edited here.

Three different MOT formats are supported. See the [mot file format](#) for more details. Depending on where the editor is launched, some formats might be unavailable.



*Airspace Activation Time Editor dialog box (stand alone mode)*

## The Airspace Activation Table

The column **zone** is the name of the sector where the Airspace Activation Time is applied. **Start Time** and **End Time** columns define the time range of the Airspace Activation.

A zone can be associated to several time period or flight level range. To do that, use the action "Split this Airspace Activation Time" in the contextual menu.

Other columns can be displayed depending of the MOT format contained in the table:

### Format 2 (time and level)

Two columns, **Lower FL** and **Upper FL** are added. They represent the flight level range where the mot is applied. Their values are always between 5FL and 595FL and can be incremented by 10FL. On loading the values are rounded to 5FL.

### Format 3 (time and type)


The column **Type** is added. It can contain the values **E** for Entry or **EX** for Entry/Exit.

## Menu

### File

- **New...** - Creates a new empty MOT file.
- **Open Mot...** - Opens a Mot file (\*.mot) and a sector file if it hasn't been previously loaded.
- **Open Sector & Mot...** - Select a sector file to load (\*.sls or \*.gsl) then select a Mot file open for edition (\*.mot).
- **Save** - Saves the Mot file
- **Save As...** - Saves the Mot file as a different name

### Edit

- **Add a new Activation Time** - Adds a new row in the Airspace Activation Time table. The zone is automatically set to the first available sector. Start time is set to 00:00 and end time is set to 24:00. Lower and upper FL (MOT format 2) are set to the values of the selected sector. The type (MOT format 3) is set to "EX"
- **Remove selection** - Removes all the selected rows. You can also remove one row by clicking on  in the last column of the table.
- **Duplicate current Activation Time** - Duplicate this row in order to specify another time period or flight level range for the sector of the current Activation Time.

### Format

- **Convert selection to MOT format 1 (only time)** - Converts the selected rows in the MOT format 1.
- **Convert selection to MOT format 2 (time and level)** - Converts the selected rows in the MOT format 2.
- **Convert selection to MOT format 3 (time and type)** - Converts the selected rows in the MOT format 3.
- **Convert All to MOT format 1 (only time)** - Converts the all the rows in the MOT format 1.
- **Convert All to MOT format 2 (time and level)** - Converts the all the rows in the MOT format 2.
- **Convert All to MOT format 3 (time and type)** - Converts the all the rows in the MOT format 3.

## Contextual Menu


- **Edit current cell** - Edit the value of the current cell.
- **Duplicate current Activation Time** - See Edit menu.
- **Add a new Activation Time** - See Edit menu.

- **Remove selection** - See Edit menu
- **Convert selection to MOT format 1 (only time)** - Converts the selected rows in the MOT format 1.
- **Convert selection to MOT format 2 (time and level)** - Converts the selected rows in the MOT format 2.
- **Convert selection to MOT format 3 (time and type)** - Converts the selected rows in the MOT format 3.

## Editing the table

Click on the **"Add a new Activation Time"** button to add an Activation Time on a sector which is not currently used in the table. The new row is appended at the end of the table. You can click on the header in order to sort the table on any column. If every sector of the dataset has been assigned with an Activation Time, this button is disabled.

If you need to specify few time periods or flight level ranges of the same sector, select the row you want to duplicated then click on **"Duplicate current Activation Time"** from the *edit* or the *context* menu.

To remove an Activation Time, click on the  button located at the right its row. You can also select all the Activation Times you want to remove and select **"Remove selection"** from the *edit* or the *context* menu.

Double click, press F2 or select **"Edit current cell"** from the context menu in a field to edit its value. The editor vary depending of the column:

- **Zone** - The list of available sector appear in a drop-down list. Only the sectors contained in the loaded sector file (sls or gsl) and which are not used in the Activation Time table appear in this list. It is not possible to add new sectors. To do so, you have to edit the sector file with the [graphical airspace editor](#).
- **Start Time and End Time** - Select a time between 00:00 and 24:00.
- **Lower FL and Upper FL** - The minimum and the maximum values are determined by the associated zone. The arrows allows incrementing and decrementing the value of 10FL.
- **Type** - Choose a value between E and EX from the drop-down list.

*Note:* When a field is in edit mode, "tab" jumps the cursor to the next field, and "shift tab" to the previous field.

## Inconsistencies

When an invalid MOT file is loaded, the inconsistencies will appear in red in the table. A tooltip helps to determine the cause of the inconsistency.

	Zone	Start Time	End Time	Lower FL	Upper FL	
1	AZ	00:00	00:00	435 FL	265 FL	⊖
2	BIRDFIS	00:00	21:59	-	-	⊖
3	EB	00:00	00:00	-	-	⊖
4	ED	00:00	24:00	-	-	⊖

*Example of inconsistencies in the Airspace Activation editor*

## 4.12 Configuration Editor

### Accessed by

Tool bar Icon 

### Purpose/description

View and edit complex airspace volumes such as ATC *Centers*, along with their sector configurations, capacities and opening schemes.

### Data model

The config editor uses the CFMU data model:

- One or several Air Blocks plus an altitude/flight level band parameter describe one ATC sector, called an **Elementary Sector (ES)**.
- Two or more ESs are "bandboxed" to **Collapsed Sectors (CSs)**.
- ESs can make up the area of an **ATC Unit Airspace (AUA)** aka Center
- An ES, CS can all have a Capacity Value.
- Several AUAs assembled together form one **ATC Units Airspaces Grouping (AUAG)**.
- The ES and CS of one AUA are used to build the Sector **Configurations** and their related **Opening schemes** (Activation table). The Opening Scheme shows the operational configuration at a given moment in time during the day

### Typical work flow

**Warning:** since the Configuration editor is an application outside the main Saam application, the workflow between the Airspace editor and the config editor might appear cumbersome. In practice you should:

1. Create, modify your airblocks and sectors with the Airspace editor in Saam;
2. Save your data in gar/gls files;
3. Launch the Configuration editor, and open the gar/gls files you just created/modified
4. Build your collapse sectors, configuration and opening schemes in the config editor.

5. Save you airspace dataset (you can keep the Config editor open)
6. Launch analysis processes in Saam.

## Input

The configuration editor works with *airspace datasets* in a Gasel format. An airspace Gasel dataset is composed of the following files:

Extension	Contents
.gar	Airblocks
.gsl	Sectors (elementary)
.spc	Collapsed sectors, ATC centers
.cfg	Configurations
.cos	Opening schemes
.cap	Sector capacity data (SAAM format)

**Warning:** the sector capacity data (.cap file) must be in SAAM format and not Gasel format.

Note: Only the Ariblocks and the Sectors files are mandatory to run the Configuration Editor.

Note: The configuration editor does not support the SAAM are/sls format.

## Output

A complete airspace dataset.

## Limitations

1. You cannot visualize graphically (in the Saam main window) the selected item of the config editor tree. And vice-versa.
2. The search function only operates on element IDs, not on the description filed of the elements.
3. You cannot create a new AUAG or modify the set of centers composing an AUAG. AUAG for the time being should be regarded as read only data.

### 4.12.1 Main window : tree view

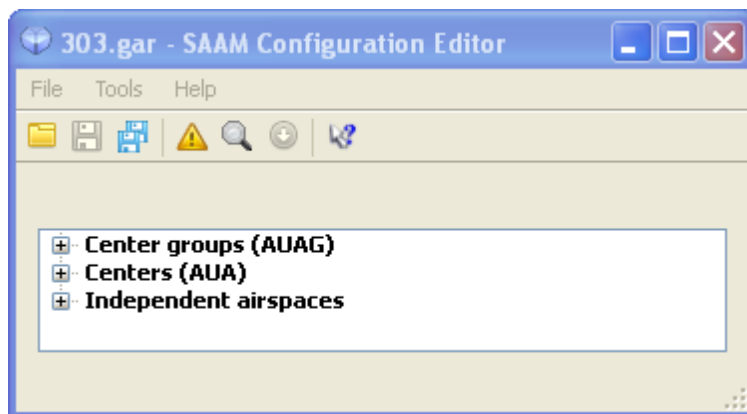
#### Treeview browser

The treeview occupies the largest part of the window and displays all objects in the dataset, in a hierarchical fashion.

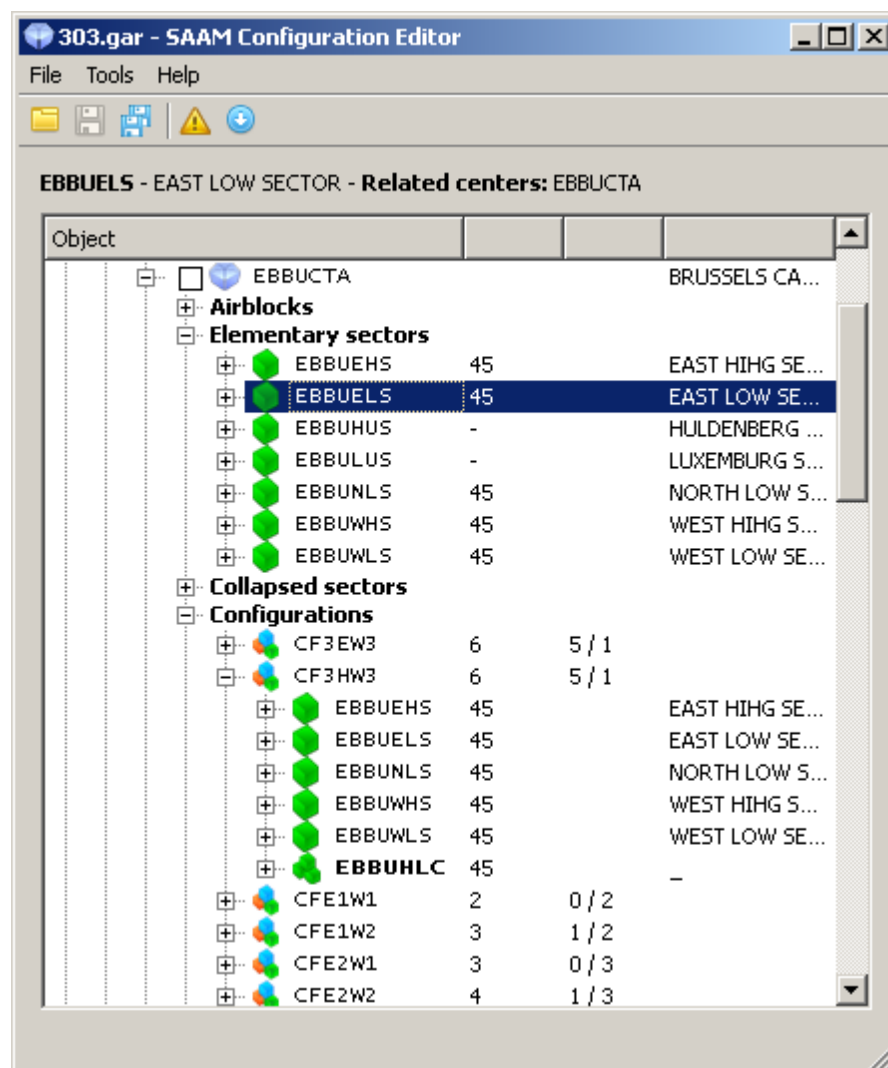
Airspace elements are grouped in 3 top level groups:

- **Center groups (AUAG):** collection of Centers e.g. defining the 8.33 core area
- **Centers (AUA):** ATC centers.

- **Independent airspaces:** elements in the dataset that do not belong to the two previous group.



Typical view when tree items are open:



## Information zone

This zone, located on top of the treeview, displays textual information related to the currently selected item. Nature of the information that is displayed depends on the type of the selected item.










## Menu bar

The menu bar contains the following menu's:

- **File** menu:
  - *Open airspace*: Closes the current dataset (if any), and loads another one.
  - *Save airspace*: Saves the current dataset. This command is available only the user has modified the dataset.
  - *Save airspace as*: Saves the current dataset to a new set of data files with different names or in a different directory.
  - *Exit*: Closes the current dataset (if any), and close the Configuration Editor. If the dataset has been modified, the user is invited to save it before exiting.
- **Tools** menu:
  - *Log viewer*: Displays the Log viewer window, containing status messages concerning the dataset loads (info, warnings, errors).
  - *Search*: search elements in your airspace
- **Help** menu:
  - *Online help*: Displays this help.
  - *What's This*: contextual help
  - *About SAAM Configuration Editor*: Displays the version of the Config editor.

## Tool bar

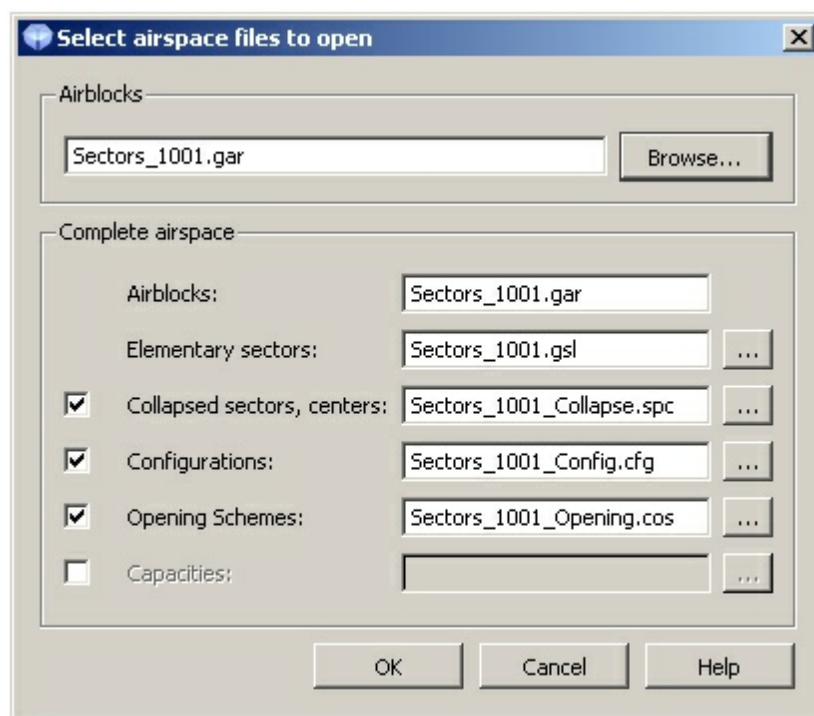
The toolbar on the Configuration Editor's main window gives quick access to the following commands:

Tool	Command
	Open airspace
	Save airspace
	Save airspace as
	Display Log Viewer
	Search items
	Uncheck all ATC centers in the treeview
	What's This help

## 4.12.2 Loading/saving airspaces

### Open airspace

The *Open airspace* command displays the dialog box represented below:



Use the *Browse* button to select your Airblocks file on disk (.gar file). The application will automatically open the other related airspace files with the same base name (not considering the extension). You can also manually select each file one by one.

It is possible to disable one of the optional airspace file by unselecting it with its associated checkbox.

Click on *OK* to open the selected files and load their contents into the Configuration Editor. The [Log Viewer](#) will report any WARNING or ERRORS found in the dataset.

#### Notes:

- The initial directory of *Browse* is your SAAM project working directory.
- At minimum, the dataset must contain airblocks and elementary sectors (.gar and .gsl files).

### Save airspace

The *Save airspace* command writes the current state of the opened airspace onto the disk.

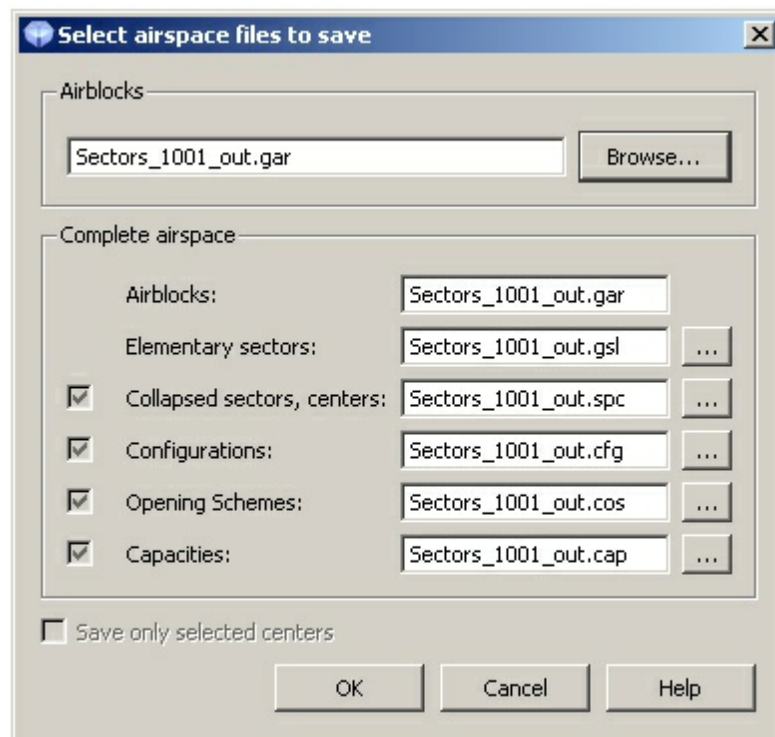
#### Note:

- [uncoherent datasets](#) cannot be saved (the *Save airspace* command is greyed out)
- if you have not modified anything to the dataset, the *Save airspace* command is greyed out.

## Save airspace as

The *Save airspace* command writes the current state of the opened airspace into a specified fileset and makes this new dataset the current one in the tree.

This dialog box is displayed:



The *Save only selected centers* checkbox allows to save only a subset of the current airspace (that is, selected centers and all items related to them). If it is unchecked, the complete airspace data is saved.

Use the *Browse button* to give a new name to the dataset files and/or specify another target directory. By default the target directory is the same as the directory from where you open the dataset.

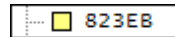
### Note:

- like the *Save* function, the *Save as* creates a complete set of airspace files, even if the original dataset was not complete. Some airspace files might be empty if certain element types are not present in the configuration.
- [uncoherent datasets](#) cannot be *Save as* (the *Save airspace as* command is greyed out)

### 4.12.3 Airblocks

#### Airblocks

Airblocks are identified by their name in the treeview:



**Note:** airblocks are created and edited using [SAAM airspace editor](#), not the Configuration Editor, which merely displays them.












### 4.12.4 Elementary sectors

#### Overview

*Elementary sector* items are created and edited using [SAAM airspace editor](#), not the Configuration Editor, which merely displays them and allows to use them as building blocks for other items. One exception: the capacity of a sector is defined/modified in the Config Editor.

#### Representation

Elementary sectors are represented in the treeview browser according to the format presented below:

	EBBUEHS	45		EAST HING SECTOR
	072EH	195	245	
	075EH	195	245	
	076EH	195	245	
	077EH	195	245	
	078EH	185	245	
	081EH	195	245	
	087EH	195	245	
	804EB	185	245	
	807EB	185	245	
	862EB	185	245	

The first line contains the name of the sector, its capacity and its description.

The child items correspond to the 3D-airblocks composing the elementary sector. Each of them is represented by its name, bottom FL and top FL.

#### View sector capacity

The capacity of a sector is displayed in the treeview and in detail in a separate window.

#### Sector capacity in the tree view

The treeview shows the default capacity and the minimum and maximum capacity if the sector has variable capacities during the 24h span of a day. In the example below, LFEEE has a capacity of 26 for the entire 24H, LFEEKE capacity is unknown, LFEESE has a default capacity of 29, and a minimum capacity of 20 and a maximum capacity of 40.

Elementary sectors		
+	LFEEE	26
+	LFEEKE	-
+	LFEEKF	-
+	LFEEKH	-
+	LFEESE	29 (20,41)
+	LFEEUE	35
+	LFEEUF	-
+	LFEEUH	38

### Detailed sector capacity in a separate window

To view all the variable capacities of a sector, right-click on it in the treeview browser, and select “*Edit capacity...*”. This will display the following dialog box:

Start time	End time	Capacity
00:00	08:00	45 (Default Capacity)
08:00	13:00	50
13:00	15:30	55
15:30	23:59	45 (Default Capacity)

The default capacity is displayed at the top of the window.

The *Variable capacity* frame presents the evolution of the sector capacity during the 24-hour span of a day. Each period is represented by its start time, end time, and the capacity of the sector during that period.

Greyed lines correspond to periods where no explicit capacity has been defined, leading to the use of the default capacity figure.

**Tip:** the capacity window is modeless: you can leave it open and navigate from sector to sector in the treeview: the capacity values in the capacity window are updated automatically.

### Modify sector capacity

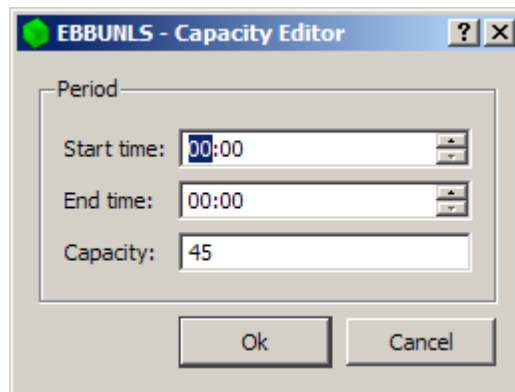
To modify the capacities of a sector, right-click on it in the treeview browser, and select “*Edit capacity...*”.

### Default capacity edition

The default capacity of the sector can be changed by typing a value in the corresponding input box and clicking on *Apply*. Numerical values in the range [0, 999] are accepted. The “-” (minus) symbol is also accepted, meaning “no default capacity defined”.

### Variable capacity insertion

Clicking on the *Insert* button opens this dialog box:



When clicking on “OK” in this dialog box, a new period with corresponding values will be inserted in the 24-hour span, automatically reducing or deleting the previously defined periods it overlaps.

### Period modification

Selecting a particular period in the table and then clicking on the “Modify” button opens the *Period insertion* dialog box, with values pre-filled. The behaviour of the dialog box is the same as above.

### Period deletion

Selecting a particular period in the table and then clicking on the “Delete” button will reset the selected period to the default capacity, eventually merging with adjacent periods.







### Remarks

Adjacent periods having the same capacity are automatically merged.

## 4.12.5 Collapsed sectors

### Representation

Collapsed sectors are represented in the treeview browser according to the format presented below:

		<b>EBBUEEC</b>	45 (10,10)	—
		EBBUEHS	45	EAST HIGH SECTOR
		EBBUELS	45	EAST LOW SECTOR

The first line contains the name of the sector, its capacity and its description.

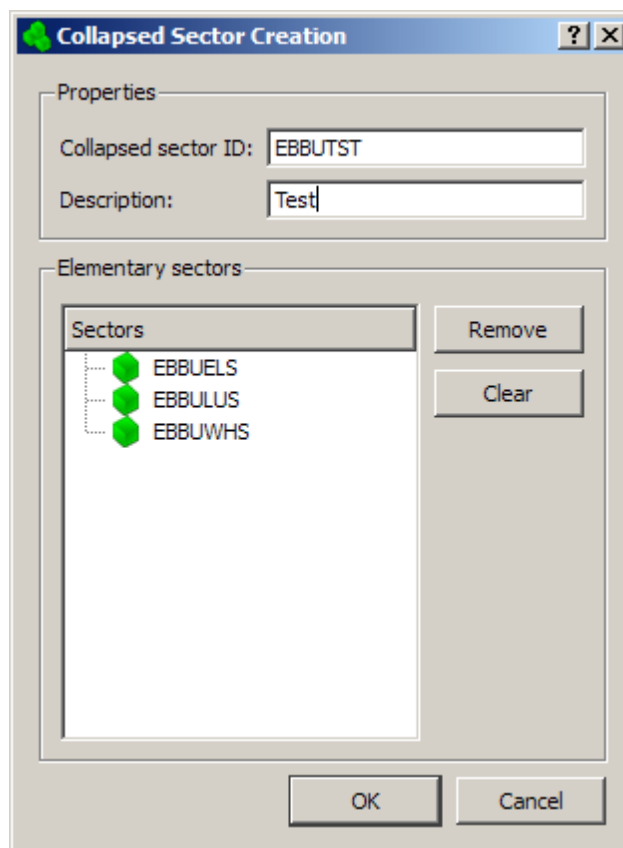
The child items correspond to the elementary sectors composing the represented collapsed sector.

### Collapsed sector creation

The “Create collapsed sector” command is available from the following locations:

- Any collapsed sector already created and present in the treeview browser.
- The “Collapsed sector” header inside any center.
- The “Unassigned sectors” header at the top level of the treeview browser.

From any of these points, choosing “Create collapsed sector” from the associated contextual menu will bring the following dialog box:



The dialog box is titled "Collapsed Sector Creation" and contains two main sections: "Properties" and "Elementary sectors".

**Properties:**

- Collapsed sector ID:** A text field containing "EBBUTST".
- Description:** A text field containing "Test".

**Elementary sectors:**

- A list box titled "Sectors" containing three items: EBBUELS, EBBULUS, and EBBUWHS, each preceded by a green circle icon.
- Buttons "Remove" and "Clear" are located to the right of the list box.

At the bottom of the dialog are "OK" and "Cancel" buttons.

The “Properties” frame allows to edit the ID and description of the new collapsed sector.

The “Elementary sectors” frame allows to define the sectors that will enter in the composition of the new collapsed sector.

To add an existing elementary sector in the new collapsed sector, follow these steps:

1. Find the elementary sector you want to add in the treeview browser.
2. Right-click on it and choose “Add to collapsed sector”.
3. The elementary sector now appears in the “Collapsed sector creation” dialog box.
4. Repeat operation for each elementary sector that needs to be added to the collapsed sector.

It is possible to remove an elementary sector from the list or to clear all sectors by clicking on the corresponding buttons.

Clicking on “OK” will effectively create the new collapsed sector. It will then be assigned to any center that includes all of its elementary sectors, or, if no center satisfies that condition, to the “Independent airspaces” category.

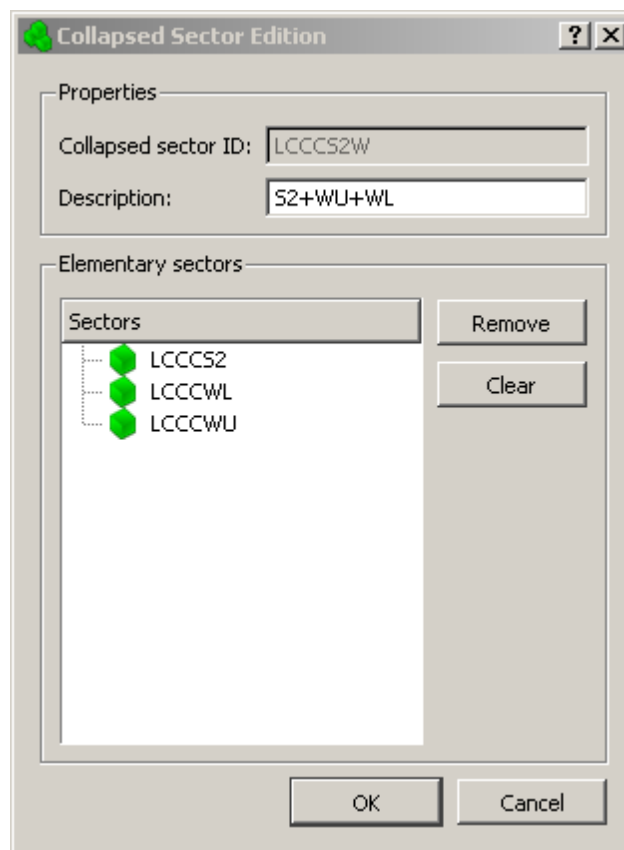
## **Remark**

It is also possible to add a collapsed sector in the composition of another collapsed sector. It will appear as a collapsed sector in the dialog box but no link will exist between the two sectors: the elementary sectors composing the first one will merely be copied into the second one.

## **Collapsed sector edition**

It is possible to edit a collapsed sector by right-clicking on it in the treeview browser and choosing “Edit collapsed sector” in the contextual menu will bring a dialog box pre-filled with data corresponding to the current state of the sector.:





Each of them is represented as described in [Elementary sector](#) topic. It is possible to change the description and the sector composition of the sector, but not its name.

Clicking on “OK” will effectively save the new state of the collapsed sector. It will then possibly be assigned to another center or to the “Unassigned sectors” category, depending on its new sector composition.

## Sector capacity viewing and editing

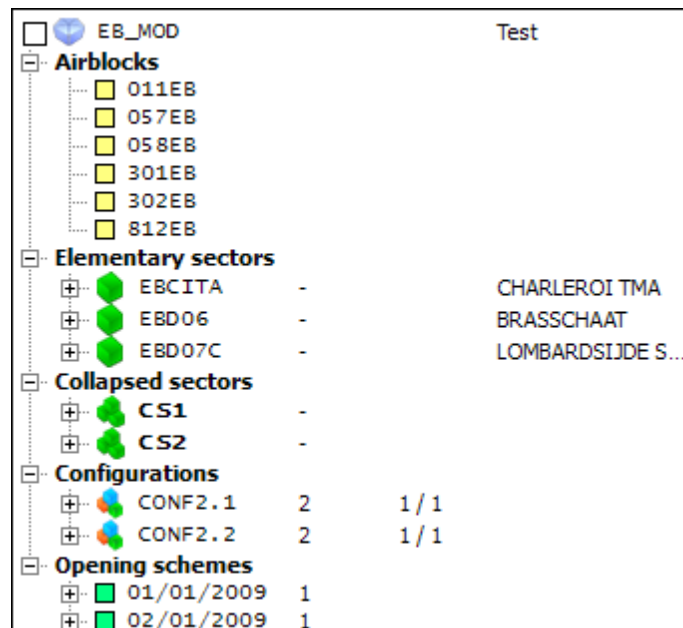
Capacities of collapsed sectors can be edited the same way as the elementary sectors (see [Capacity viewing and editing of Elementary sectors](#)) .

### 4.12.6 Centers (AUA)

#### Representation

Centers (AUA - ATC Unit Airspace) are represented in the treeview browser according to the format presented below. A center is identified by its name and a description. All items belonging or related to the center are represented as child items, grouped by category:

- [Airblocks](#)
- [Elementary sectors](#)
- [Collapsed sectors](#)
- [Configurations](#)
- [Opening schemes](#)



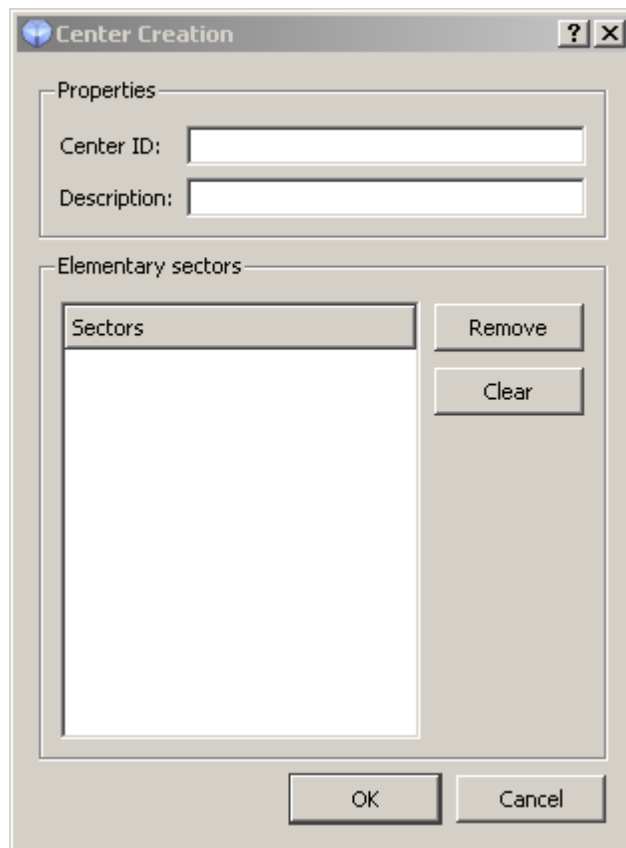
Each center has a checkbox located next to its name. This is used to “mark” a center, in order for example to perform a “Save as” operation only on the marked center.

## Center creation

The “Create center” command is available from the following locations:

- Any center already created and present in the treeview browser.
- The “Centers (AUA)” header at the top level of the treeview browser.

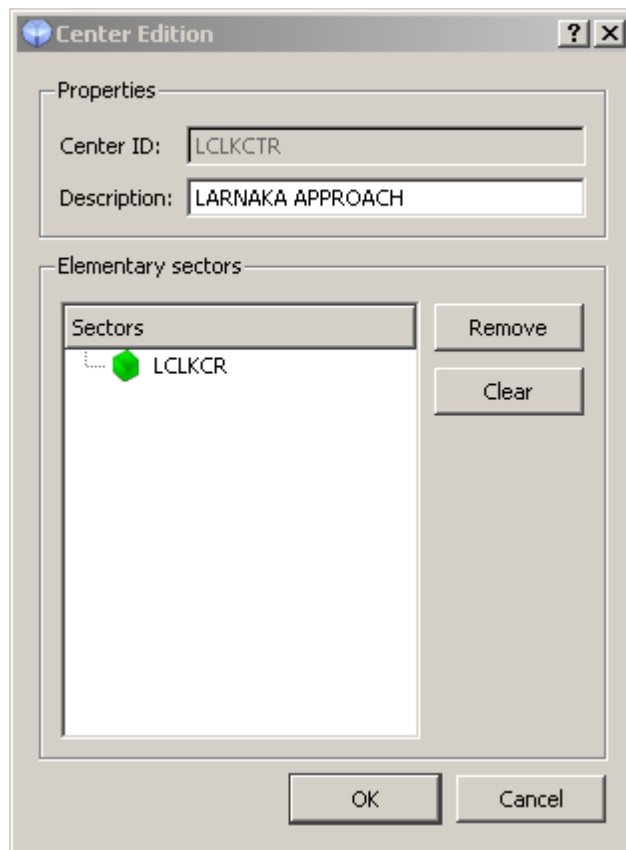
From any of these points, choosing “Create center” from the associated contextual menu will bring the following dialog box.



After creation, the new center will be placed under the appropriate “State” header, according to the two first letters of its ID.

## Center edition

It is possible to edit a center by right-clicking on it in the treeview browser and choosing “Edit center” in the contextual menu. This will display the following dialog box, pre-filled with data corresponding to the current state of the center:



It is possible to change the description and the sector composition of the center, but not its name.

Clicking on “OK” will effectively save the new state of the center. Some elementary and/or collapsed sectors can be transferred from the center to the “Independent airspaces” category (or inversely), depending on the new sector composition of the center.

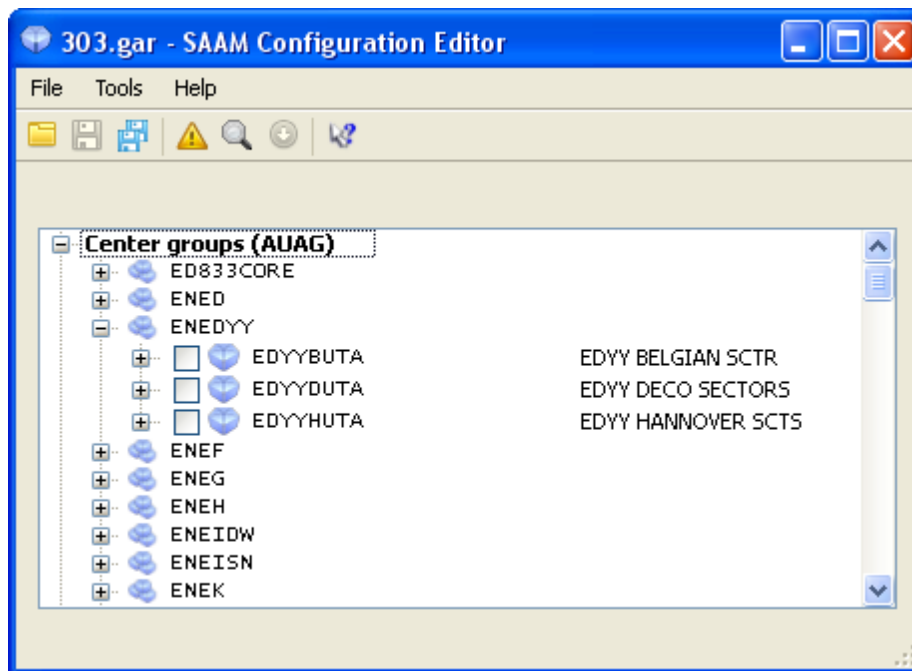
## See also

- [Save airspace as.](#)
- [Collapsed sector creation.](#)

### 4.12.7 Center Groups (AUAG)

Several AUAs assembled together form one **ATC Units Airspaces Grouping (AUAG)**.

AUAGs are grouped in the **Center groups (AUAG)** top level section of the tree view.



### Warning & limitations

- You cannot create a new AUAG.
- You cannot modify the set of centers composing an AUAG.
- You can create a new center but the center will not be attached to the current group.





















You can modified objects of a center in a AUAG but this way of working is probably not natural and you should do it from the Centers (AUA) section of the tree.

In conclusion, you should regard the AUAG objects as read-only. Most probably if you have AUAG in your dataset, the definitions originates from the CFMU database (via the Gasel files).

## 4.12.8 Configurations

### Representation

Centers are represented in the treeview browser according to the format presented below:

		CFE2W3	5	3 / 2	
		EBBUNLS	45		NORTH LOW SECTOR
		EBBUWHS	45		WEST HIGH SECTOR
		EBBUWLS	45		WEST LOW SECTOR
		EBBUEEC	45 (10,10)		-
		EBBUEHS	45		EAST HIGH SECTOR
		EBBUELS	45		EAST LOW SECTOR
		EBBUHLC	45		-
		EBBUHUS	-		HULDENBERG SECTOR
		EBBULUS	-		LUXEMBURG SECTOR

The first line contains: the name of the configuration, the number of sectors, elementary sectors / collapsed sectors in it.

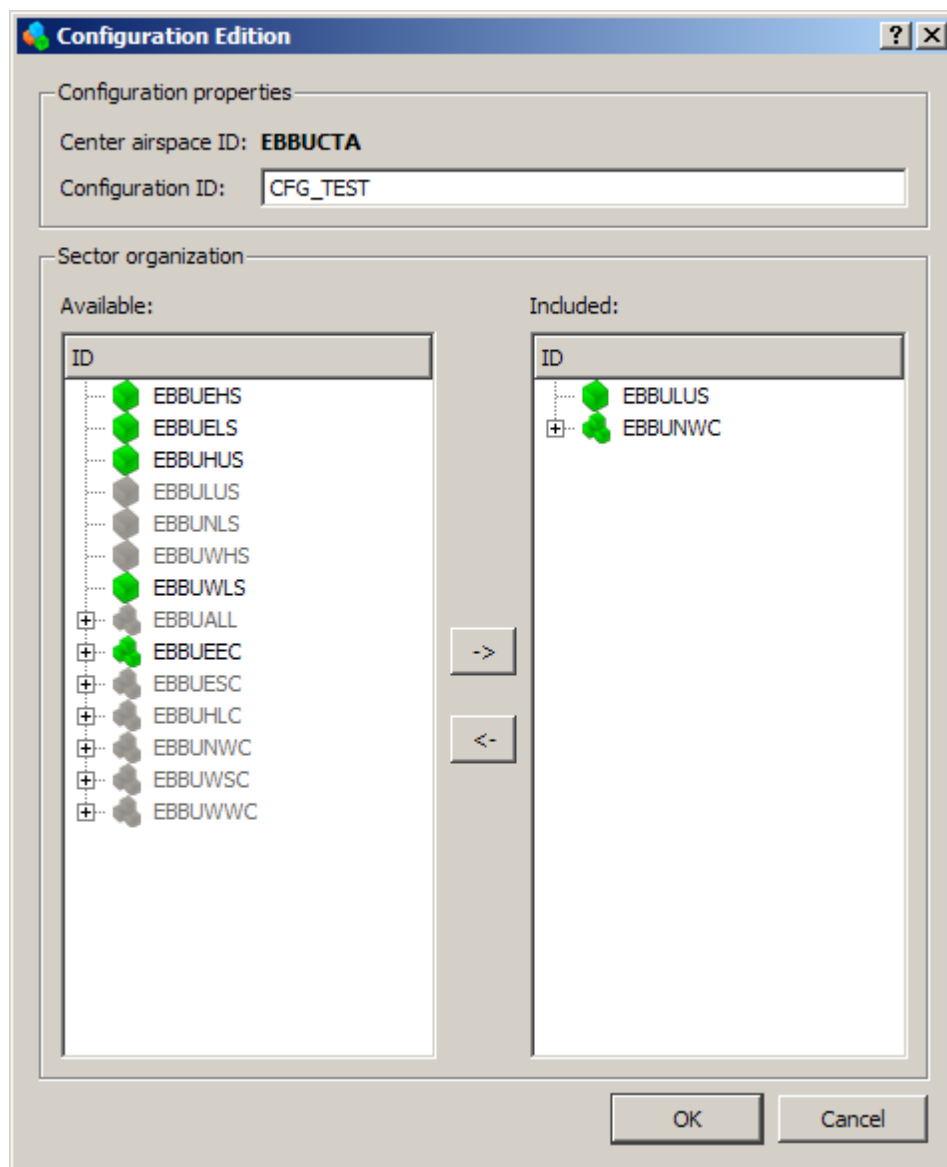
The child items correspond to the sectors (elementary and collapsed) composing the configuration. Each of them is represented as described in [Elementary sector](#) and [Collapsed sectors](#) topics.

### Configuration Creation (Ctrl-N)

The “Create configuration” command is available in the right click menu in the treeview of the following items:

- configurations in the treeview.
- centers in the treeview .
- The “Configurations” header inside any center.

From any of these points, choosing “Create configuration” from the associated contextual menu will bring the following dialog box:



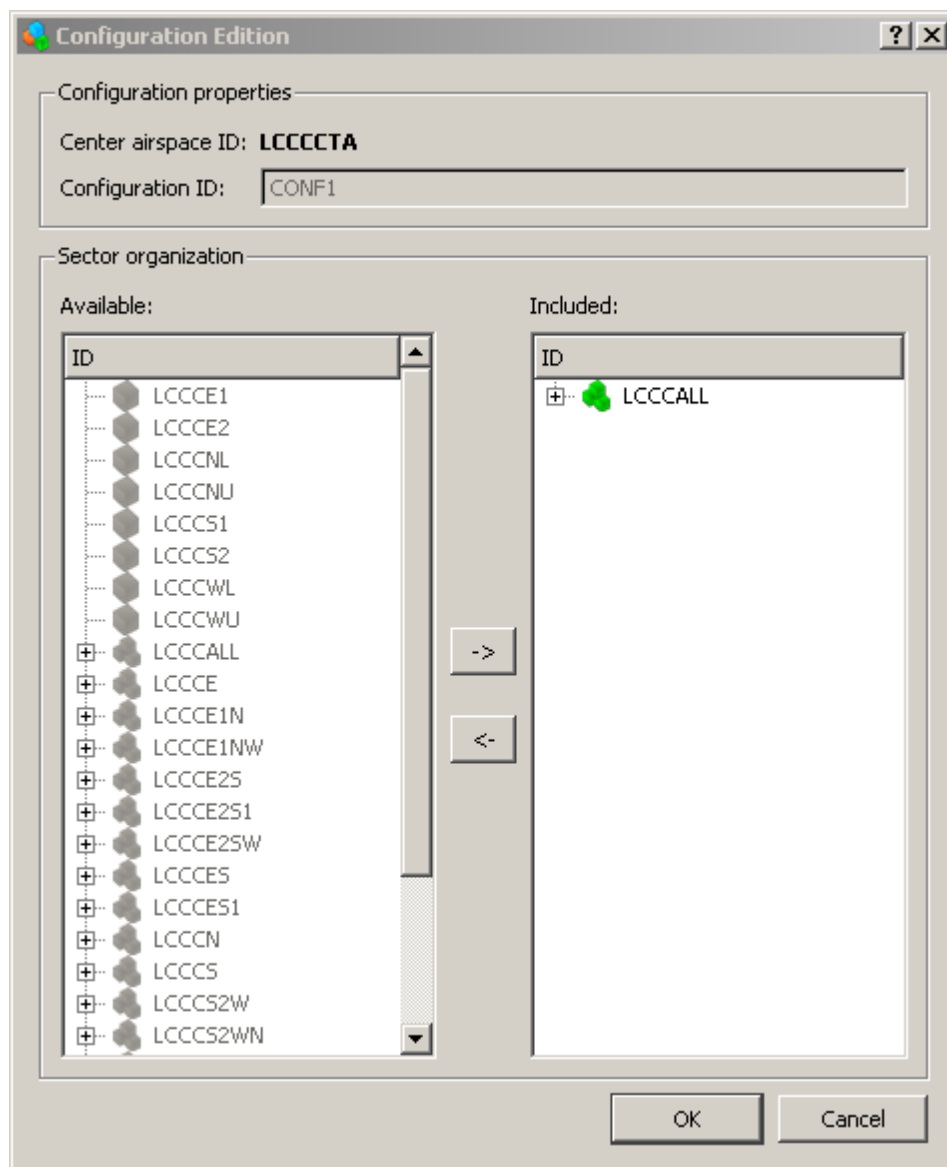
**“Create configuration” Dialog Box**

Features of the dialog box:

- ID of the configuration is to be typed in the edit box at the top of the window.
- Left part of the window presents all items available in the selected center. Only the items that are available to be included in the new configuration are coloured, otherwise they are greyed out.
- Right part of the window presents all items already included in the configuration.
- Items can be transferred from one side to the other using the “arrows” buttons in the middle of the window.
- Clicking on “OK” checks the configuration and creates it if it is valid.

## Configuration Edition (to change its composition)

It is possible to change the composition of a configuration by right-clicking on it in the treeview browser and choosing “Edit configuration” in the contextual menu. This will display a dialog box pre-filled with data corresponding to the current state of the configuration.:



Clicking on “OK” will effectively save the new state of the configuration.

## Rename a configuration (F2)

To rename a configuration, select *Rename Configuration* in the contextual menu of the configuration.

## Duplicate a configuration (Ctrl-D)

To duplicate a configuration, select *Duplicate Configuration* in the contextual menu of the configuration. A copy of the original configuration will be created with the extension *\_copy*

## Delete a configuration (Del)

To delete a configuration, select *Delete Configuration* in the contextual menu of the configuration. If the configuration is used in an opening scheme, the system will refuse to delete the item. Please remove it the opening scheme(s) first.



## 4.12.9 Opening schemes

### Representation

Opening schemes are represented in the treeview browser according to the format presented below:

02/01/2008	9	
CFE1W1	00:00	04:59
CF3HW3	05:00	05:29
CFE4W3	05:30	07:29
CF3EW3	07:30	09:59
CFE2W3	10:00	12:59
EBBUNLS	45	NORTH LOW SECTOR
EBBUWHS	45	WEST HIGH SECTOR
EBBUWLS	45	WEST LOW SECTOR
EBBUEEC	45 (10,10)	-
EBBUHLC	45	-
CF3EW3	13:00	14:59
CFE4W3	15:00	17:19
CF3EW3	17:20	19:29
CFE2W2	19:30	23:59

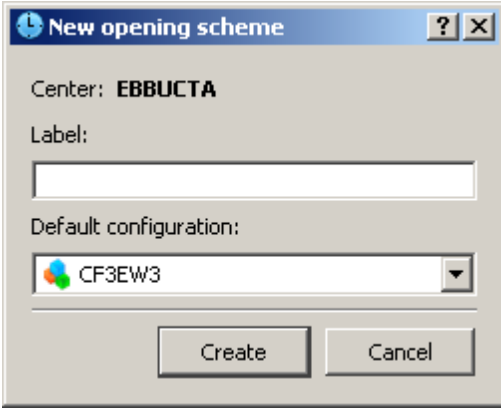
The parent item presents the name of the opening scheme (usually a date) and the number of periods (or *timeslices*) it contains. All these timeslices are represented as child items, with the name of the configuration used, start time, and end time.

### Opening scheme creation

Creating a new opening scheme is possible from the following locations in the treeview:

- a center in the treeview.
- an opening scheme in the treeview.
- The “Opening schemes” header inside any center.

From any of these points, choosing “Create opening scheme” from the associated contextual menu will bring this dialog box:



The dialog box titled "New opening scheme" contains the following fields and controls:

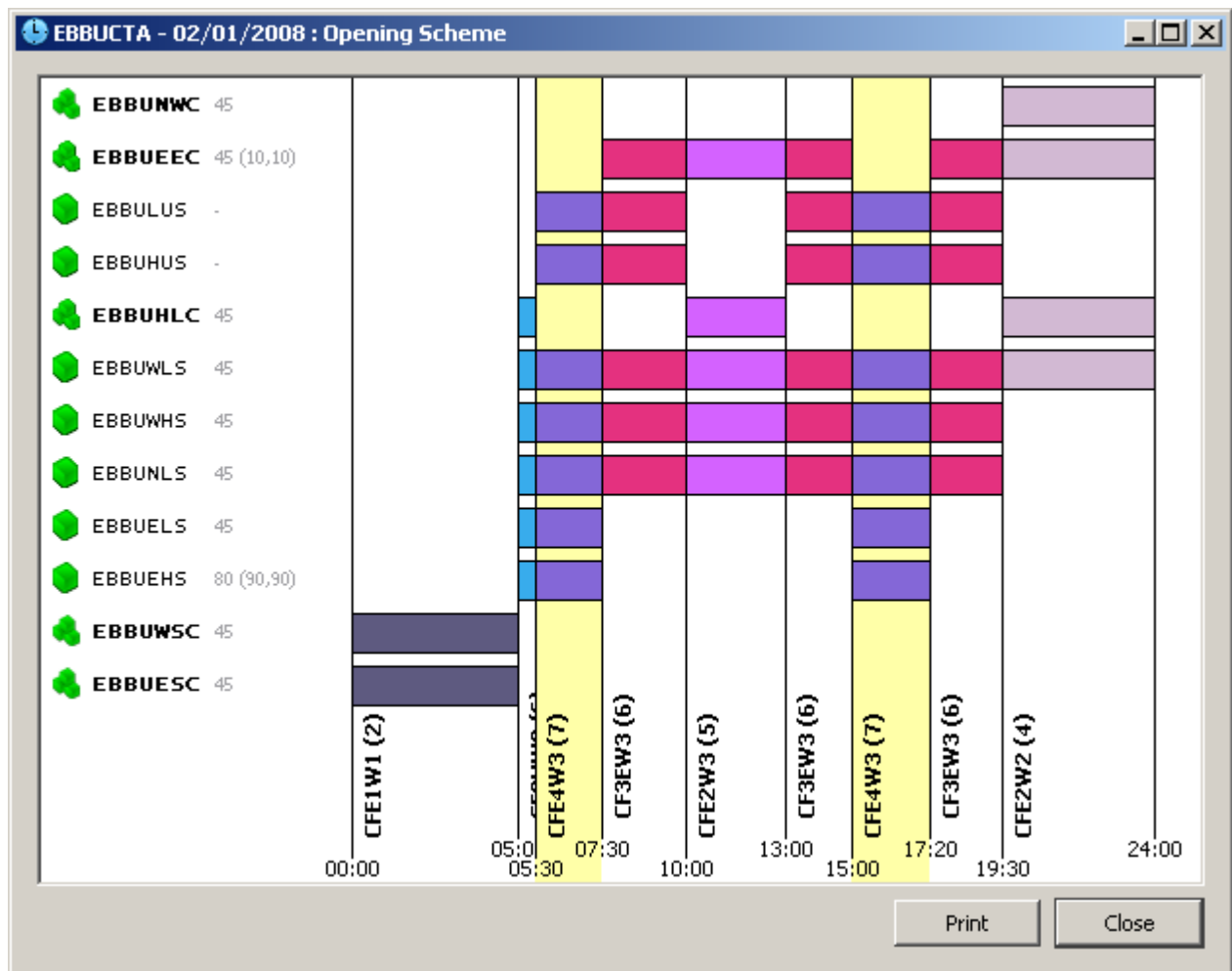
- Center:** EBBUCTA
- Label:** (empty text box)
- Default configuration:** CF3EW3 (selected from a dropdown menu)
- Buttons:** Create, Cancel

Enter the name of the new opening scheme and choose a configuration among those

belonging to the selected center. A new opening scheme will be created inside that center, using the selected configuration from 00:00 to 23:59. To add additional configurations, edit the newly created opening scheme.

## Opening scheme graphical viewer

Right-click on any opening scheme in the treeview browser and select “View opening scheme” to open a window displaying it on the complete 24-hour span, with configurations used during each timeslice and their associated sectors:



Scheme viewing

The viewer will be refreshed each time another opening scheme is selected in the treeview browser. **Tip:** you can browse the treeview with the Up/Down arrow keys.

## Opening scheme edition

Right-clicking on any opening scheme in the treeview browser will display a contextual menu, containing the following actions:

- **Delete**
- **Replace by**
- **Insert before**
- **Insert after**

Remark: Replace and Insert actions are accompanied by a sub-menu containing all available

configurations for the current ATC center.

### **Timeslice deletion**

This action will remove the selected timeslice from the opening scheme. The timeslice taking place just before the deleted one will be extended in a way to leave no gap in the opening scheme. In the case where the first timeslice (beginning at 00:00) is deleted, the timeslice taking place just after the deleted one will be extended.

### **Timeslice substitution**

This action will replace the configuration currently used in the selected timeslice by the configuration selected in the contextual sub-menu.

### **Timeslice insertion**

This action will insert a new timeslice before (resp. after) the selected timeslice. The new timeslice will use the configuration selected in the contextual sub-menu. The selected timeslice shall be reduced to one half, to clear room for the new timeslice. All other timeslices shall not be modified.

### **Automatic timeslice merging**

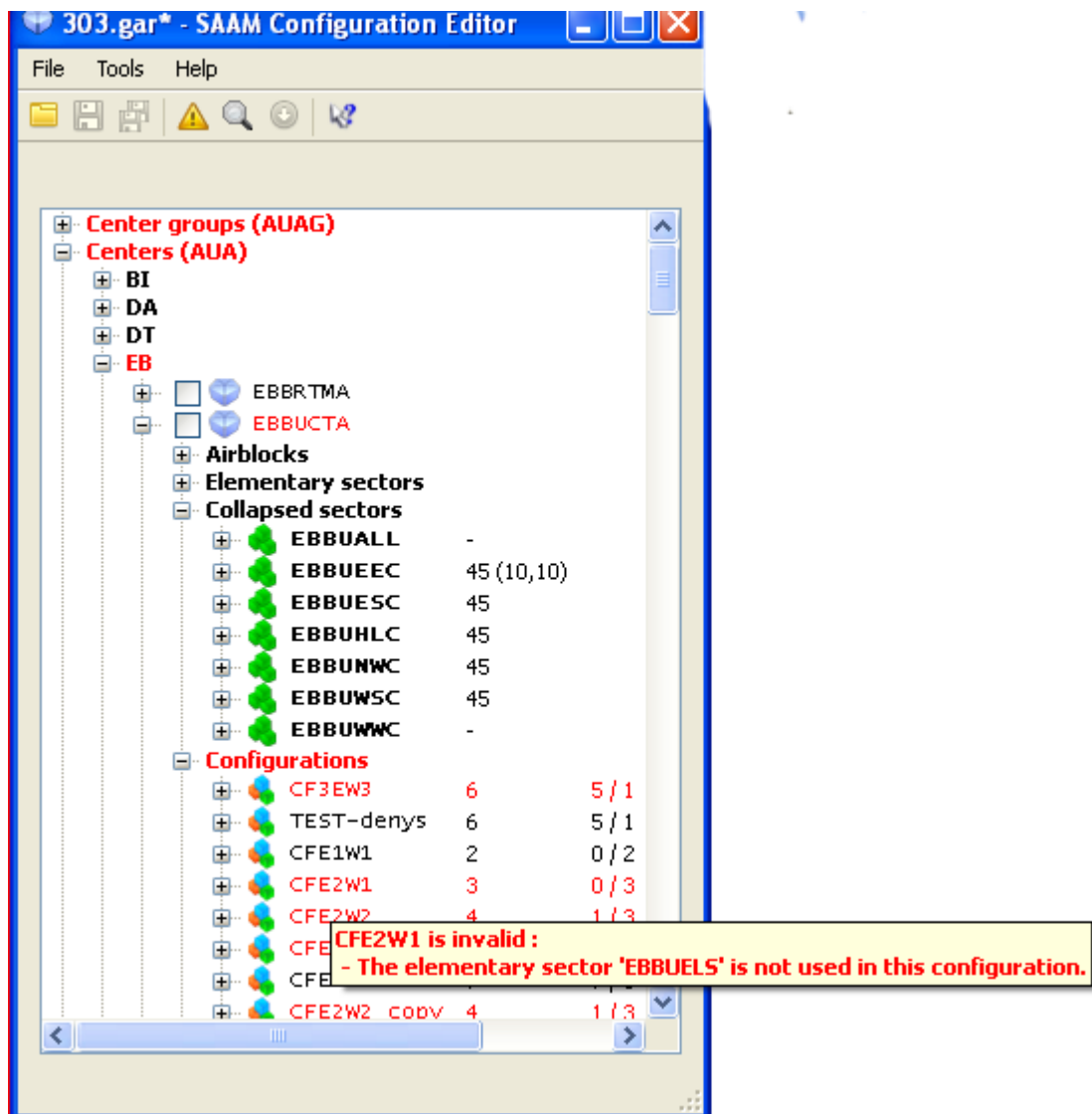
If, after a data modification (deletion, substitution, insertion), two or more adjacent timeslices use the same configuration, they shall be merged automatically.

## **4.12.10 Incoherent airspace dataset**

The configuration editor allows you to delete most items freely. In certain circumstances, this will make the airspace dataset incoherent (e.g. if you remove a sector used in a configuration).

With this approach you can make a set of related modifications/deletion in a step by step approach. While you are doing these editions, the dataset is unstable. No problem. Normally at the end of your modifications, your tree/dataset should be in a coherent state again. In any case, the application will not allow you to save an unstable dataset.

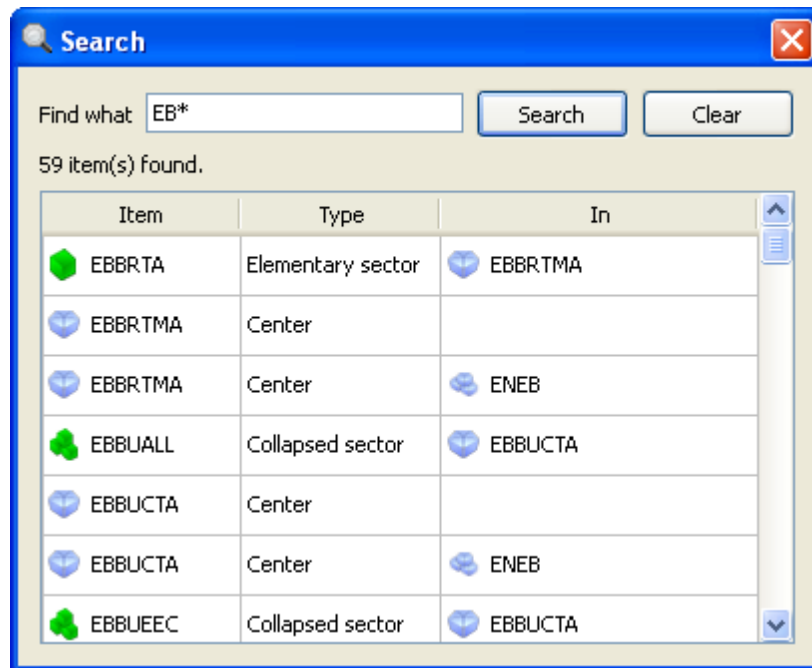
When an element becomes unstable (e.g. because it is based on another item that you have deleted), the application highlights it in red. If you hover over it with your mouse, the application shows a tool tip explaining why this item is unstable. See the example below:



### 4.12.11 Search

You can search the airspace dataset for items matching a given search criteria. For the moment the search is limited to the ID of the items.

To launch the search, click Ctrl-F or select Search in the Tools menu. The following window will appear:



### Search criteria

In the *Find what* box, enter a string for an exact search or a string with wildcards for pattern search matching. Basic wildcards are:

- \* represents zero or more characters.
- ? represents exactly one character:

*Examples:*

- **EBBRTMA**: all items whose ID = EBBRTMA
- **E\*TMA**: TMA in countries starting with E
- **\*TMA**: all TMAs (or at least all the items with an ID ending with TMA)
- **17???**: all IDs starting with 17 and containing exactly 5 characters.
- **E\*B?**: all IDs starting with E and ending with B followed by exactly one character.

The search is case insensitive: you can type EBBUCTA or EbbUCTA indifferently.

Click on the *Search button* to launch the search.

Click on *Clear* to clear the current search.

**Note:** more advanced regular expressions are accepted. As an example you can use a search criteria such as **E[0-9]\*** meaning any ID starting with letter E (or e) followed by a digit followed by zero or more characters. The description of these advanced regular expressions are out of scope of this help. Please refer to information found in Wikipedia.

## Search results list

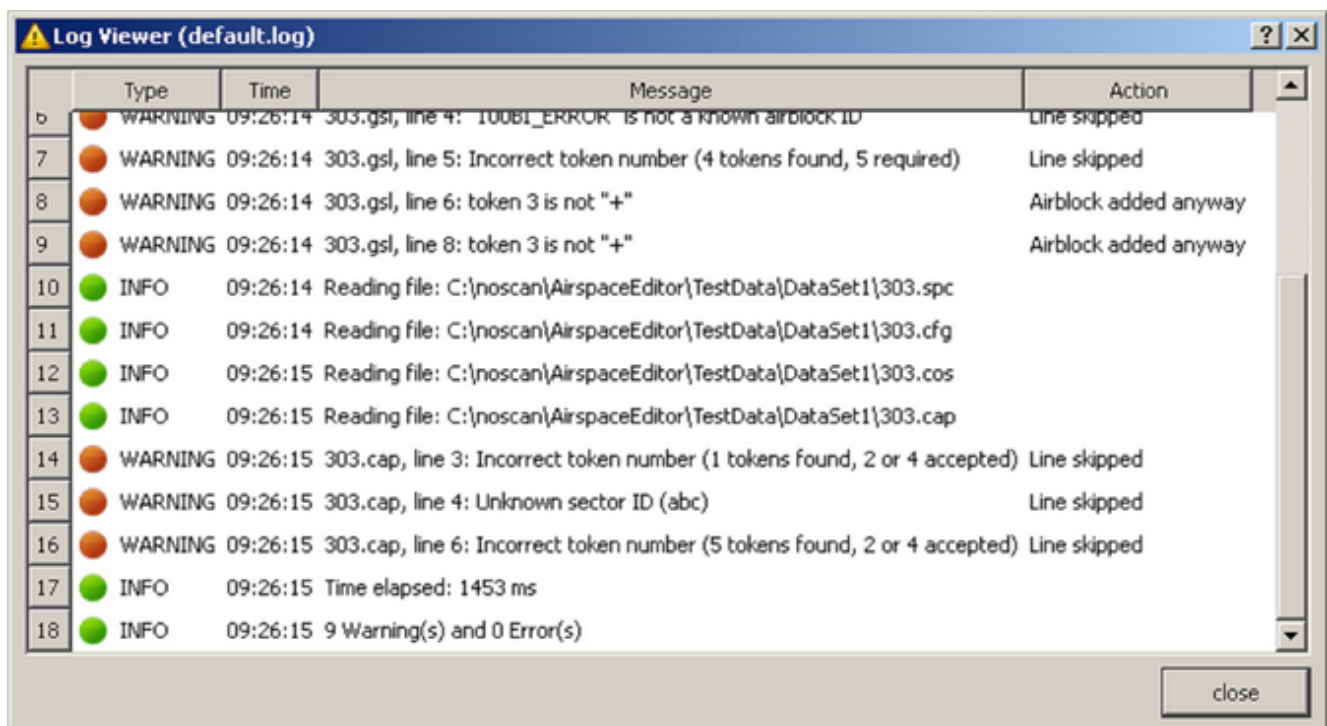
The search result lists contain the list of all items whose ID meets the search criteria. In addition, the list indicates the type of element and the containing center if the item is not a center.

## View an item in the treeview

To highlight in the treeview an item of the results list, click on the item in the result list: the element will be highlighted in the treeview in grey. If necessary, the application will automatically open the necessary hierarchy to view the item.

### 4.12.12 Log viewer

The *Log Viewer* window collects the status messages emitted throughout the Configuration editor's airspace dataset loading process:




Three categories of messages are tracked:

- **INFO**: Purely informative message, for example, indicating which file has been loaded.
- **WARNING**: Message indicating that some non-blocking error occurred. In some cases the Configuration Editor is able to perform an action to repair that error; this action is then displayed in the "Action" column of the Log viewer.
- **ERROR**: Message indicating that a blocking error occurred. The dataset is rejected altogether.

**Note:** the data of the log viewer is stored in a text file called **default.log** located in the same folder as your current TDV. Please send this file to the Saam support team in case you have a problem with a dataset.

## 4.13 Rule Editor

### Accessed by

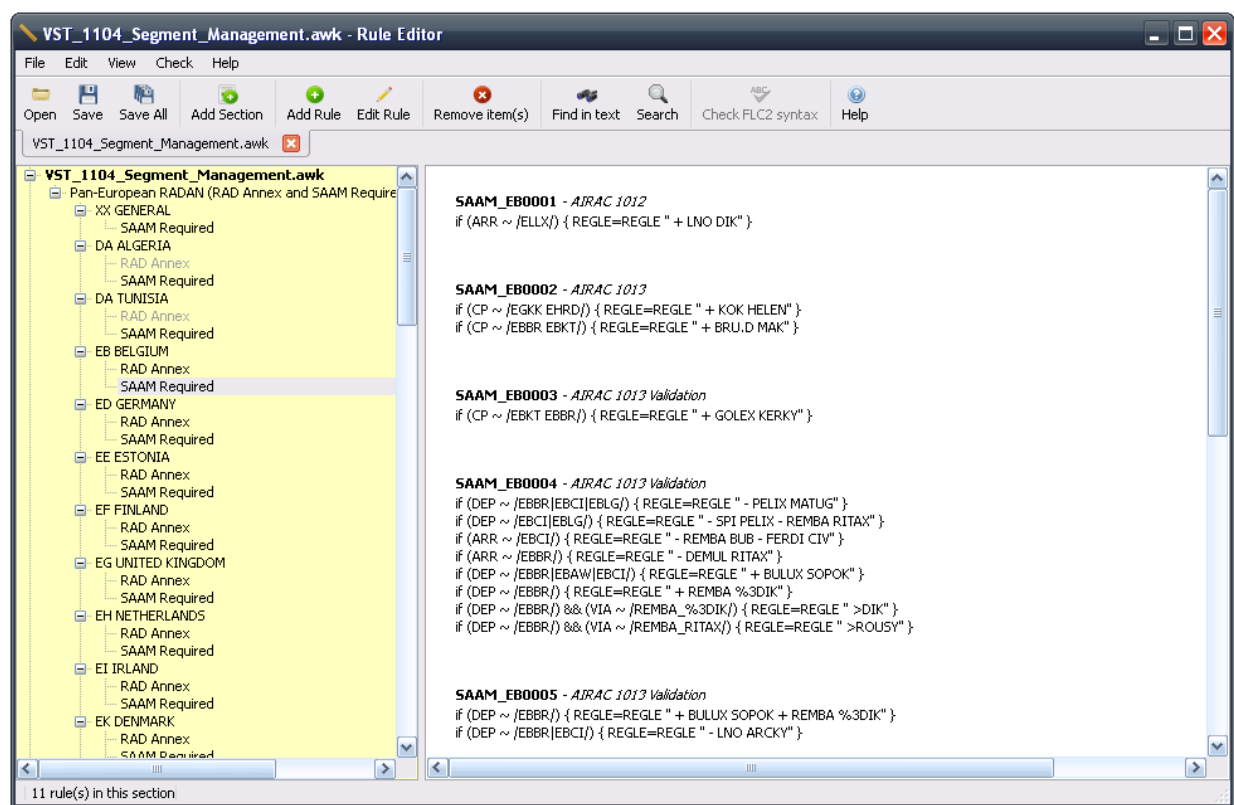
The editor can be activated by clicking on the  button in the SAAM toolbar.

### Purpose/description

The rule editor allow viewing and editing rule files, Segment Management Rules (\*.awk) and Flight Level Constraints (\*.flc2) files. It is fully integrated in the layer engine environment.

### Organization

The rules are organized in sections. A section can contain rules and subsections. They are presented in a tree on the left side of the editor. Clicking on a section display the rules it contains on the left side. The sections which doesn't contain anything appear in grey.



*The main window of the Rule Editor*

### Menu bar

The menu bar contains the following menu's:












- **File menu:**

- **New** (Ctrl+N): Creates a new empty rule file.
- **Open** (Ctrl+O): Opens an existing rule file from the disk or from the current layer.
- **Save** (Ctrl+S): Saves current file.
- **Save As** (Ctrl+Alt+S): Saves the current tab in a new file.
- **Save All** (Ctrl+Shift+S): Saves all opened files.
- **Print** (Ctrl+P): Prints content of the current files.
- **Print Section**: Prints selected section.
- **Close** (Alt+F4): Closes the editor. If one tab has been modified but not saved, a popup invites the user to do it before exiting.
- **Edit menu**:
  - **Cut rule(s)** (Ctrl+X): Copies and removes selected rules.
  - **Copy rule(s)** (Ctrl+C): Copies selected rules.
  - **Paste rule(s)** (Ctrl+V): Pastes copied rules in the selected section.
  - **Add Section**: Adds a new section at the top level of the tree.
  - **Rename Section**: Renames selected section.
  - **Add Rule** (+): Adds a new rule in the displayed section.
  - **Edit Rule...** (Enter): Edits selected rule.
  - **Remove Item(s)** (Del): Removes selected item (section or rule).
- **View menu**:
  - **Find in text** (Ctrl+F): Finds or replaces a keyword in the displayed rules.
  - **Search** (Ctrl+Alt+F): Searches a keyword in the complete dataset.
  - **Error list**: Displays the list of error found in the current file.
- **Check menu**:
  - **Check FLC2 syntax**: Checks the syntax of the current flight level constraint file (FLC2). This action is not available for the other types of file.
- **Help menu**:
  - **Help** (F1): Displays this help.


## Tool bar

The toolbar on the rule editor gives quick access to the following actions:

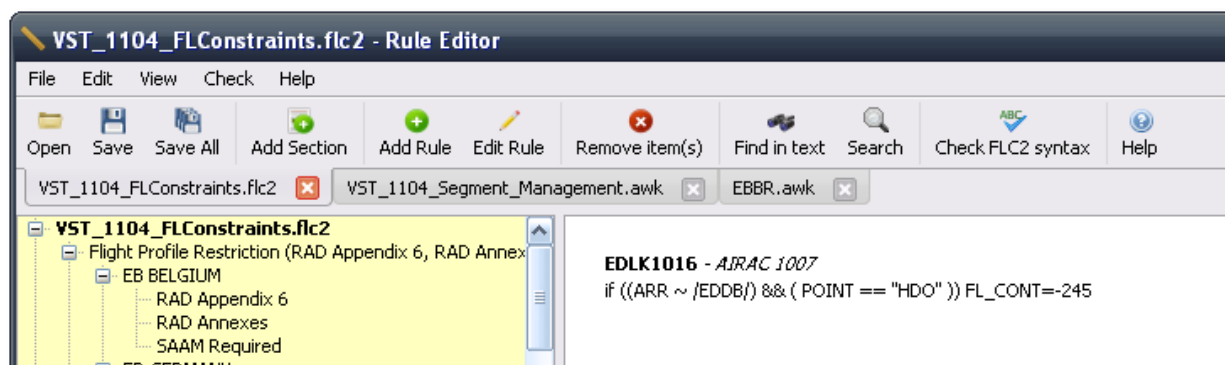


icon	Command
 <b>Open</b>	Opens an existing rule file
 <b>Save</b>	Saves current file
 <b>Save All</b>	Saves All files
 <b>Add Section</b>	Adds a new section at the top level of the tree.
 <b>Add Rule</b>	Add a new rule in the displayed section
 <b>Edit Rule</b>	Edit selected rule
 <b>Remove Item(s)</b>	Remove selected item (section or rule)
 <b>Find in text</b>	Finds or replaces a keyword in the displayed rules
 <b>Search</b>	Searches a keyword in the complete dataset
 <b>Check FLC2 syntax</b>	Check the syntax of the current file (only for FLC2 files).
 <b>Help</b>	Displays this help

## Multi tab management

After opening a file, it appear in a new tab. Several files can be opened at the same time. Click on their tab to switch between them. You can close them by clicking on the  button on each tab.

Note that when a file has been modified a '\*' appear in its tab name. If you close it, a popup will ask you if you want to save your changes.



**Tabs in the Rule Editor**



## Limitations

1. The shortcuts are not currently working.
2. Undo/Redo is not implemented. Be careful when you edit something.

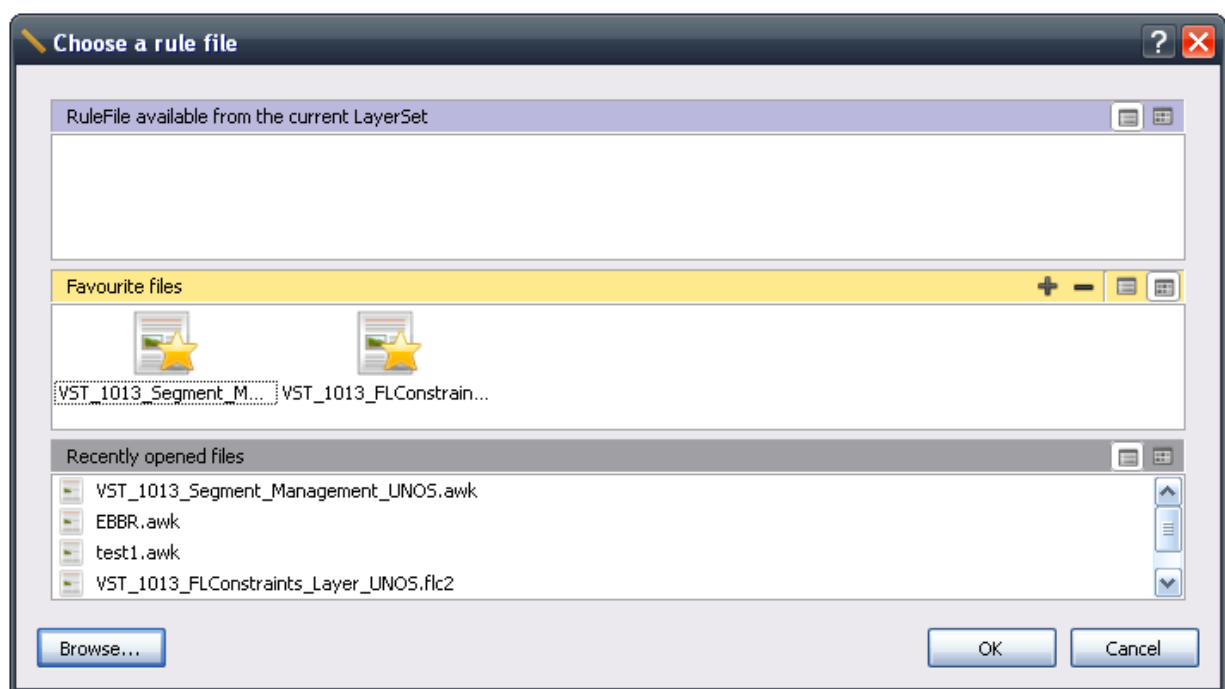
### 4.13.1 Open a rule file

#### Description

To open a rule file, you click on the **Open** button. A dialog box appears that contains three groups. These groups gives a quick access to recently opened files, favourite files and layered files. See their description below.

The **Browse...** button in the bottom left corner allow browsing a rule file, either \*.awk or \*.flc2. The presentation of each group can be switched between **icon view** and **list view** by clicking on  or .

**Tip:** All the display settings of the dialog box are stored in your computer. If you close SAAM and open it later, you will find the dialog box unchanged.



Rule editor Open dialog box

#### Rule files available from the current LayerSet

If a layer set has been opened with SAAM, the rule data associated with the layer set will appear in the top group. For more details see the [layer engine integration](#).

## Favourite files

The favourite files are files you want to have easy access to. They appear in the middle group in yellow. It is possible to add as many favourite file as necessary by clicking on the "+" button. Clicking on the "-" button removes the selected favourite file.

## Recently opened files

The five last opened files appear in this group.

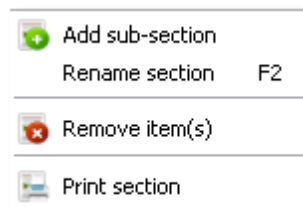
### 4.13.2 Edition

## Editing the sections

### Section context menu


Right-clicking on a section opens a context menu that gives access to the following commands:

- *Add sub-section*: Adds a sub-section in the selected section.
- *Rename section* (F2): Renames the section.
- *Remove item(s)* (Del): Removes the selected section(s).
- *Print section*: Prints the section



The section context menu

### Moving a section

It is possible to re-organize the sections by moving them with drag & drop. To do so, select a section, hold the left mouse button and move the cursor to where you want to move the section, then release the mouse button. If the icon  appear next to the mouse cursor, it means that the section cannot be moved to this location. Pressing the **Escape** button during the process will cancel the action.

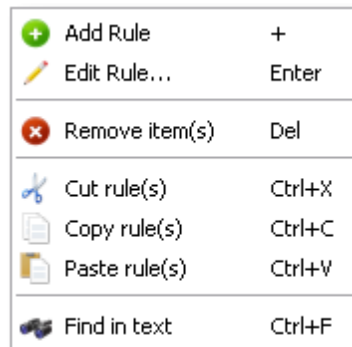
**Tip:** You can select more than one section at a time to move or delete them.

## Editing the rules

### Rule context menu

Right-clicking on a rule opens a context menu that gives access to the following commands:

- *Add Rule (+)*: Adds a new rule in the current section.
- *Edit Rule... (Enter)*: Edit the selected rule.
- *Remove item(s) (Del)*: Removes the selected rule(s).
- *Cut rule(s) (Ctrl+X)*: Copies and removes the selected rule(s).
- *Copy rule(s) (Ctrl+C)*: Copies the selected rule(s).
- *Paste rule(s) (Ctrl+V)*: Pastes the copied rule(s) in the current section.
- *Find in text (Ctrl+F)*: Find or replace a keyword in the rules of the current section.



The rule context menu

**Tip:** Moving the mouse cursor over a rule displays four buttons which give quick access to the Edit, Remove, Cut and Copy actions.

## Creating and editing a rule

When the **Add rule** or **Edit rule** command is triggered the rule edition dialog box is opened.



The rule edition dialog box

A rule is composed of three fields:

- The Id: It is mandatory and it must be unique among all the rule of the file.
- The comment: An optional description of the rule.
- The body: The definition of the rule.

Clicking the OK button will validate the data and create or update the rule. If the entered Id is not unique, a warning message asking you to select another Id will appear.

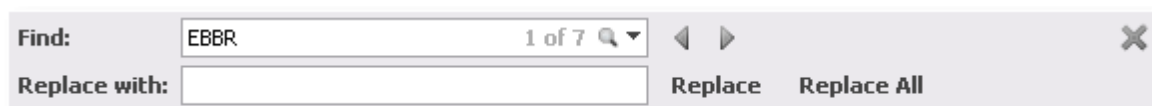
**Tips:**

- Every group of '?' inserted in the Id field will be automatically replace by a number of 4 figures which guarantee that the Id is unique. Use this function for fast Id creation.
- In the body text field, the syntax is highlighted. Comments appear in **green**, keywords in **dark blue**, operators in **brown**, quoted text in **grey**. Also the spaces are marked with a middle point.

### 4.13.3 Search

#### Find in text


Clicking on the **Find in text** command displays the following dialog box at the bottom of the editor. It allows finding a keyword in the **currently displayed** rules.

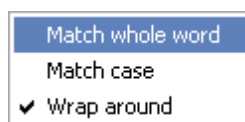


The "find in text" widget

Type your keyword in the **Find** text field. The number of match is written in grey on the right. The keywords are highlighted in green in the rules. You can switch between the results by clicking on the arrows.

To replace the keyword by another text, type the new text in the **Replace with** text field and click on **Replace** do to it one by one. Or you can click on **Replace All** to replace all the keywords in one shot.

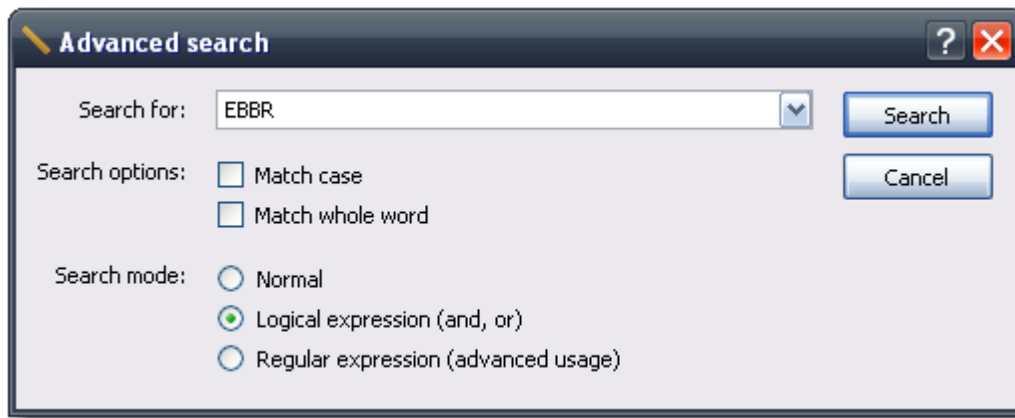
Clicking on  opens the following menu which allows modifying the options of the find.



The find menu

#### Search

The search dialog box is an advanced tool for searching one or more keywords in all rules of the file. Enter your text in the **Search for** field and click on then **Search** button.



The search dialog

**Tip:** The drop down list contains the last ten search which has been done.

### Search modes

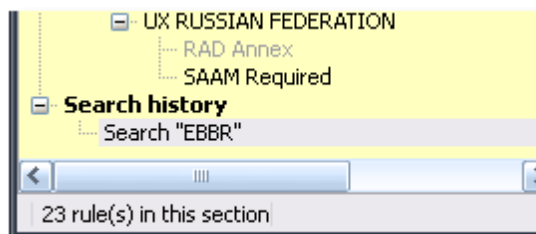
In addition of the classic search options (*match case* and *match whole word*), three search modes are available:

- *Normal*: Search the text exactly as it is written.
- *Logical expression*: Separates the keywords with the logical operators "**and**" or "**or**".  
*Examples*: The query "EBBR and EBCI" will retrieve every rules which contain the word EBBR and the word EBCI. The query "EBBR or EBCI" will retrieve the rules which contain one of the keyword.
- *Regular expression*: Handle the query as a regular expression (advance usage).

In practice, you can always use the Logical expression mode except if your query contain the words "or" or "and".

### Presentation of the results

The rules which match the search will be grouped together in a custom search section which appear at the bottom of the section tree, under the label "*Search history*".

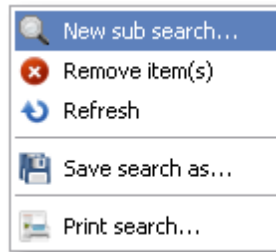


Search history

Clicking on this section display the rule that the search have found. The rules contained in the searches still belong to their sections. You can find the real location of a rule by right-clicking on it and selecting the command "Display in its original section" in the context menu. In a search section you can perform every usual edition actions like adding, editing or removing a rule. Only the "Paste" command is disabled.

The context menu of the search section is a bit different. It contains the following command:

- *New sub search...*: Performs a search based on this one.
- *Remove item(s)* (Del): Removes the search.
- *Refresh*: Refreshes the search.
- *Save search as...*: Saves the content of the search in file.
- *Print search*: Print the search.




Search section context menu

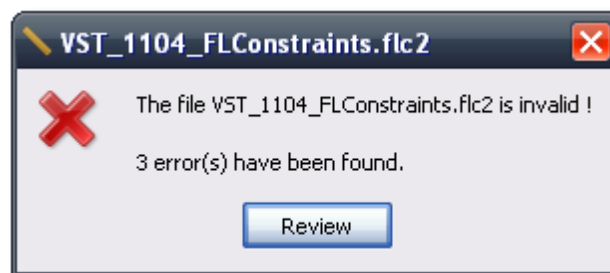
**Tip:** To replace a keyword in all the rules of the file, start by performing a search with an empty string to retrieve all the rules in one search section. Then perform a replace with the "Find in text" command.

#### 4.13.4 FLC2 Syntax checking

##### Syntax checking

A tool is available to check the syntax of any flight level constraint files (FLC2). Clicking on the  icon in the **check** menu or in the tool bar will run the process.

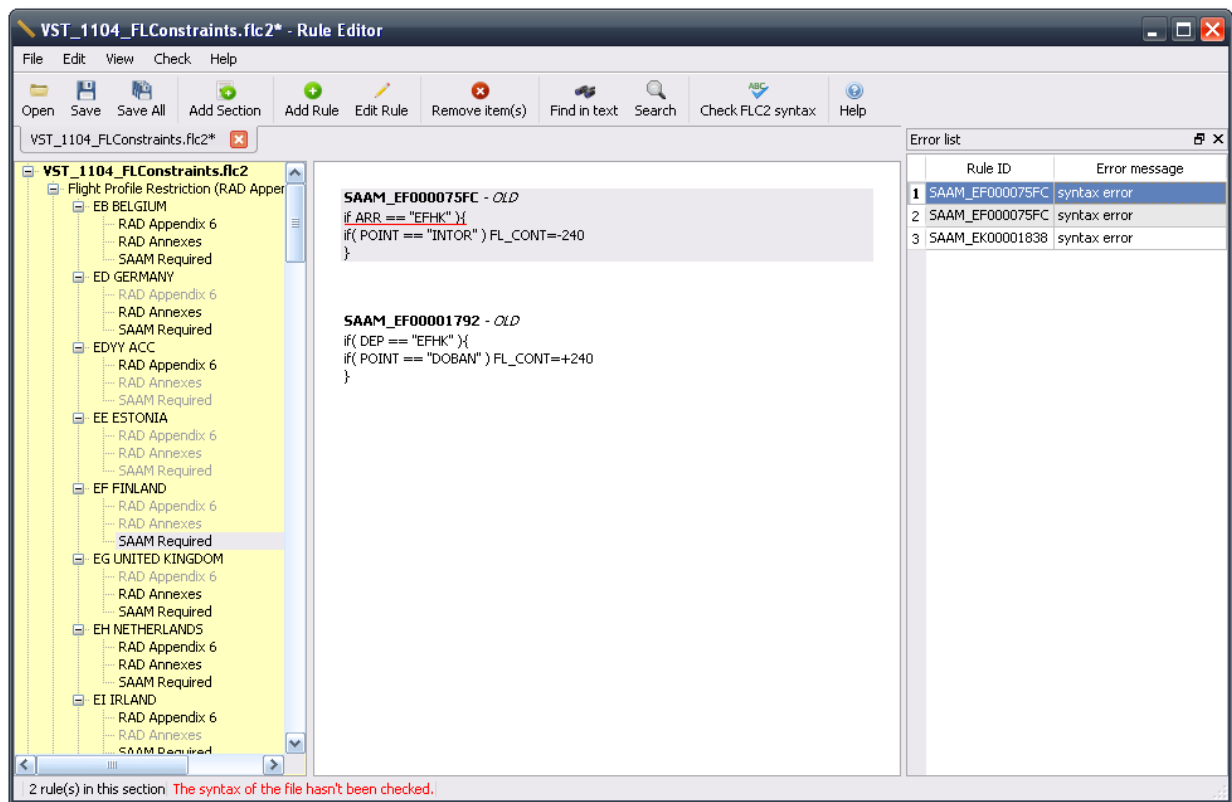
A message box will appear to give you a summary of the checking. If some errors have been detected, clicking on the **Review** button displays the detailed list of error and highlight the first one.



Syntax checking summary

##### Browsing and fixing the errors

The list of errors is displayed in the Error list view. It is automatically displayed after that an invalid syntax checking has been performed but it can also be opened and closed from the **view** menu.



The error list in the rule editor

Selecting an error from the list automatically display the associated rule. The line which contains the error is underlined in red. To fix the it, simply edit the rule and change its text in the [rule edition dialog box](#).

**Note:** The errors stay in the list even if they have been fixed. To validate you changes you need to run the checking process again.

### Tips:

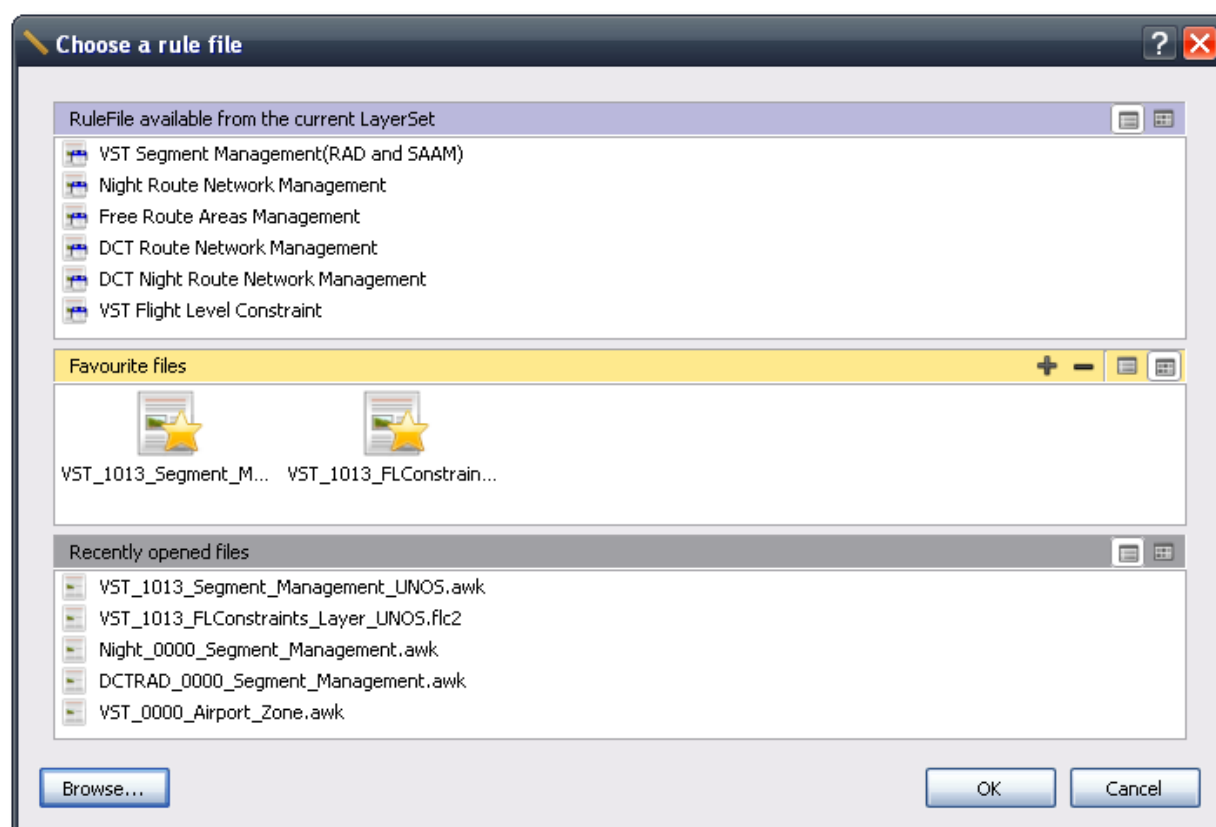
- Sometimes, a simple syntax error, like a missing parenthesis, can invalidate most of the following rules. So it is advised to fix the errors in the order they appear in the list.
- Because syntax checking can be sometimes a heavy process, it isn't done automatically. It's up to you to perform it when you need it. However, a warning message appear in red at the bottom of the window, when the file that you are editing has not been checked and/or contains errors.

## 4.13.5 Layer engine integration

### Opening a file from the current LayerSet.

The rule editor is fully integrated with the layer engine of SAAM. When a LayerSet is loaded in SAAM, the files available from this LayerSet appear in the first group of the open dialog box. Select one of them to edit it.





The open dialog box when a layer set is loaded

## Edition

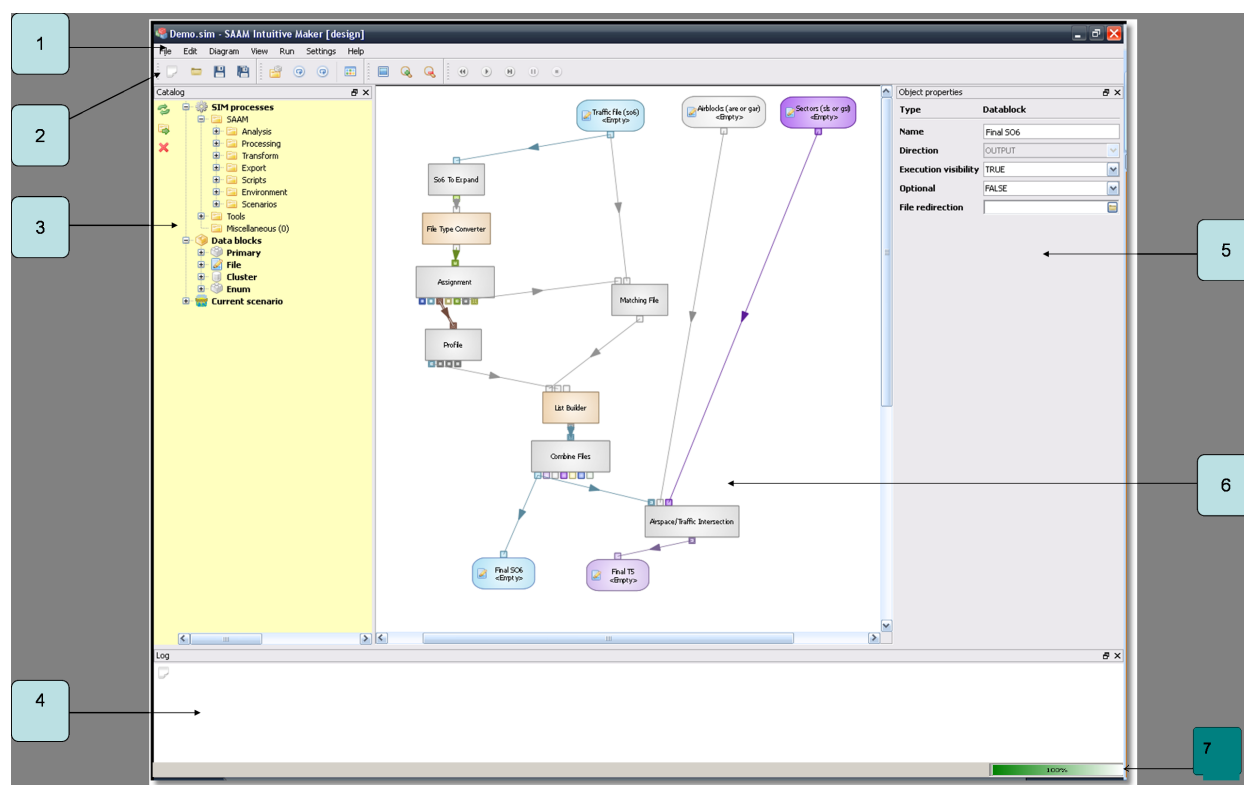
The edition works in the exact same way as a normal rule file except that the **Save** command is disabled. All your modifications are registered in the layer engine and will be saved when you click the main SAAM save button.

When the selected layer in the layer dashboard is not editable (because it is hidden or because it is an airac or the baseline), the edition action of the editor is disabled. Furthermore a warning message is displayed in the status bar.

**Tip:** You can open simultaneously several layered files in different tabs. You can even mix them with normal files.

## 4.14 SIM User Manual

### Presentation of the SIM main window



**SIM main window**

The SIM main window is composed of following parts (refers to the numbers shown in figure above):

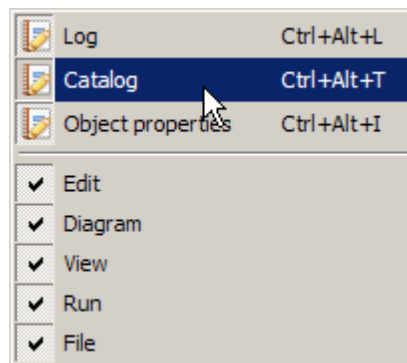
- 1. Menu bar:** Access to the complete set of features of SIM Designer.
- 2. Toolbar:** Quick access to the most commonly used features of SIM Designer.
- 3. Catalogue panel:** Floating panel containing building blocks for diagrams (processes, data blocks, etc). These elements are ready for drag-and-drop to the diagram view. Look at [this page](#) for more information about the content of the catalogue.
- 4. Log panel:** Floating panel to display log messages produced by the application.
- 5. Object properties panel:** Floating panel for viewing and editing properties of the element currently selected in the diagram view.
- 6. Diagram view:** Central zone of the main window, for viewing the currently opened diagram and interacting with it.
- 7. Progress bar:** Progression indicator for the current diagram execution.

### Floating layout

The SIM main window has a floating layout, the size and position of all panels and toolbars appearing in the main window can be changed to better suit your preferences. All elements can be shown or hidden at any time. Panels can be docked in the main window or released from it to become independent, “floating” windows. Docked panels can be organized as you

wish and even stacked on top of each other. Toolbars can be docked on any side of the main window or released from it to become independent, “floating” toolbars.

The *floating elements pop-up menu* allows showing or hiding any floating element of the SIM Designer main window:



**Floating elements pop-up menu**

This pop-up menu can be displayed by right-clicking on:

- The free space on the menu bar.
- The free space on any toolbar.
- The title bar of any **docked** panel.

Once the menu is displayed, click on any of its elements to switch the visibility of that element on or off. A hidden element will always be redisplayed having the same position and size as when it was hidden in the first place.

Any toolbar can be moved by clicking on its anchor and dragging it to the desired place in the screen.



**A toolbar and its anchor**

A docked panel has a title bar similar to the one shown in the following figure:



**Docked panel title bar**

The title bar shows the name of the panel and two buttons for undocking or hiding the panel.

You can also undock the panel by double-clicking on its title bar, or left-clicking on the title bar and dragging it to another place.

A floating (i.e. undocked) panel has a title bar.

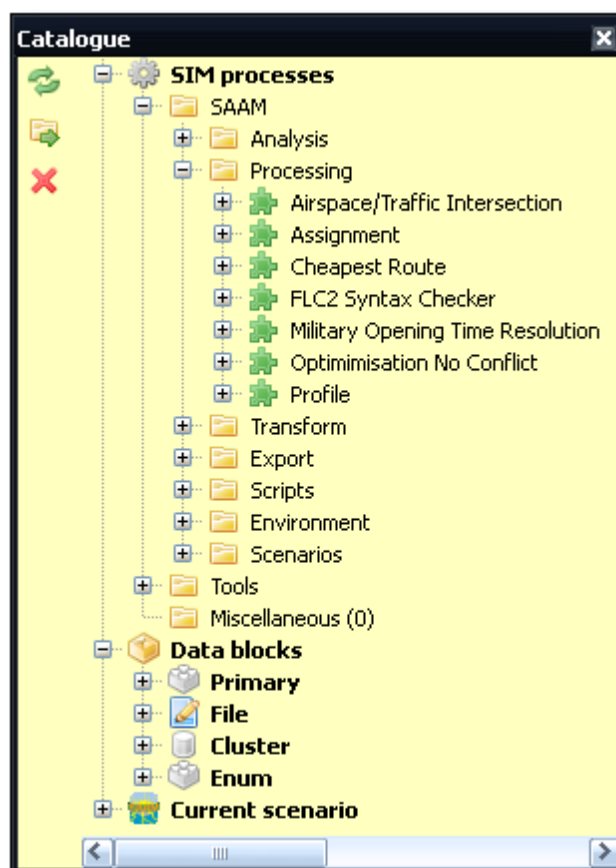


Floating panel title bar

You can dock a floating panel by double-clicking on its title bar (it will then go back to its last docked position) or by left-clicking on the title bar and dragging the panel to any side of the main window (the panel will then be docked or stacked, depending on the cursor position and the presence of other panels at that position).

Please note also that the *Catalog* and *Object properties* panels can only be docked on the left or right side of the main window. The Log panel can only be docked on the bottom side of the main window. No restriction is applied when those panels are floating, though.

### 4.14.1 The catalogue





The SIM catalogue

### Organisation of the catalogue

The SIM catalog contains all components that are necessary to create a SIM diagram. To add a component in your diagram, simply drag it from the catalogue and drop it on the diagram view.

It is divided in three main sections:

- **SIM processes**, contains all processes that can be run from SIM. This section is divided in two subsections:
  - **SAAM**, contains most of the processes you can find in SAAM, predefined scenarios and some additional SAAM-related tools.
  - **Tools**, contains miscellaneous non-business tools to help building complex diagrams.
- **Data block**, contains available input/output types. All file types supported by SIM are referenced here under the group "**File**".
- **Current scenario**, contains all SIM diagrams that you have saved in your SAAM working directory. To facilitate your work, diagrams used in your current work should be saved here.

At any moment, you can refresh the catalog by clicking on the  button in the upper left corner of the catalogue. You can also add a custom storage directory by clicking on . These directories are shared by all users of the computer. They reference any diagram you save inside and are independent of your working directory. You should use them to store your personal sub-diagrams.

To help you building your own diagrams, you may start from an existing one in the catalogue : by right-clicking on it, you can choose to directly edit the existing one (first choice) or edit a copy of it (second choice) :

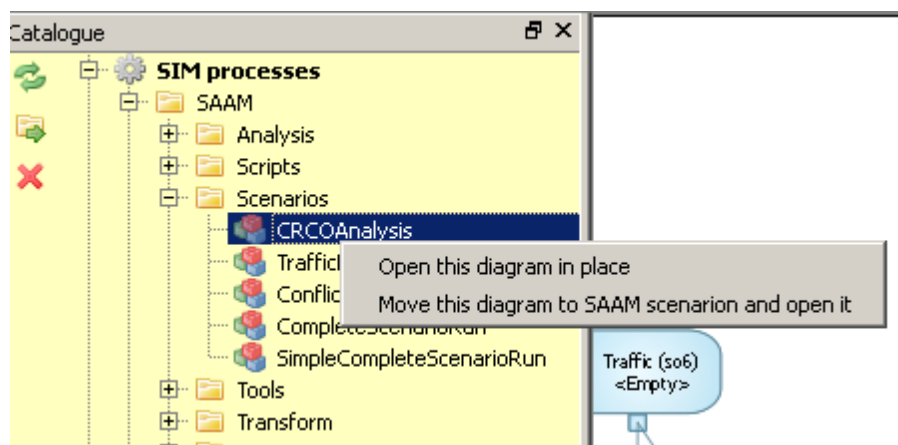


Diagram editing choice

#### 4.14.2 Menus and functions

This section lists the features of SIM accessible through its menus and toolbars.

##### Menu bar

The menu bar contains the following menu's:

- **File** menu:
  - **New** (Ctrl+N): Creates a blank new diagram in the diagram view.
  - **Open** (Ctrl+O): Opens a selected SIM file in the diagram view.
  - **Save** (Ctrl+S): Saves the diagram in the associated SIM file.





- **Save As** (Ctrl+Alt+S): For user to save the diagram under a new name and/or location.
- **Import** (Ctrl+I): Imports an exported diagram from its exported folder. Imported folder is added as a user folder.
- **Export** (Ctrl+E): Exports into a selectable folder, all what is needed to rerun the diagram : diagram, sub-diagrams, processes, plugins, parameters, input files
- **Close** (Ctrl+F4): Closes the current diagram.
- **Exit** (Ctrl+Q): Exit the SIM application.
- **Open most recent files**: The 5 most recent SIM files opened by SIM Designer are listed at the bottom of the File menu for quick access. Click on any of the files to open it.
- **Edit menu**:
  - **Select all items** (Ctrl+A): Select all items of the current diagram.
  - **Delete selected item(s)** (Del): Delete all currently selected items.
  - **Copy** (Ctrl+C): Copy the selected items in the clipboard.
  - **Paste** (Ctrl+V): Paste the previously copied items on the diagram. Pasted items are then selected and it's possible to move all of them at once.
  - **Switch data block cardinality(s)** (Ctrl+Alt+D): If the item currently selected is a data block, switches its cardinality from single to list, or inversely.
- **Diagram menu**:
  - **Add sub diagram** (Ctrl+Alt+S): Opens the selected SIM file and adds it as a sub-diagram to the current diagram.
  - **Add for loop** (Ctrl+L): Adds an empty loop structure to the current diagram.
  - **Add indexing loop** (CTRL+I): Add an empty auto-indexing loop structure to the current diagram.
  - **Properties...** (Ctrl+P): Changes the properties of the current diagram.
- **View menu**:
  - **Full view** (o): Centers the view and manipulates the zoom factor so the current diagram will be fully visible in the diagram view.
  - **Zoom in** (+): Increases the zoom factor.
  - **Zoom out** (-): Decreases the zoom factor.
  - **Panels**:
    - **Log** (CTRL+ALT+L): Show/hide the log panel.
    - **Catalog** (CTRL+ALT+T): Show/hide the catalogue panel.
    - **Object properties** (CTRL+ALT+I): Show/hide the object properties panel.
- **Run menu**:
  - **Reset**: Resets the generated output values of the diagram.
  - **Start** (F5): Starts the execution of the diagram.
  - **Step-by-step execution**: If the diagram is in design state, start the step-by-step execution. If the diagram is in paused state, resume the execution. *NB*: in both cases, only one step will be executed then the diagram will switch back to paused state.
  - **Pause**: Pauses the execution of the diagram. The currently ongoing execution step will be completed then the diagram will switch to paused state.
  - **Stop**: Stops the execution of the diagram. The currently ongoing execution step will be completed then the diagram will switch to design state.

- **Settings** menu:
  - **Preferences...**: Displays the SIM Preferences dialog box.
- **Help** menu:
  - **Help** (F1): Displays this help.
  - **About**: Displays the SIM about box.





## Tool bars

SIM interface contains four toolbars which give a quick access to the main functions. Each toolbar is independent and can be detach from its original location. You can show/hide any toolbar and panels by right-clicking on a toolbar or on a menu.




### The File toolbar

icon	Command
 <b>New</b>	Creates a new blank diagram and displays it in the diagram view.
 <b>Open</b>	Opens a selected SIM file in the diagram view..
 <b>Save</b>	Saves the diagram in the associated SIM file.
 <b>Save As</b>	For user to save the diagram under a new name and/or location.





### The Diagram toolbar

icon	Command
 <b>Add sub diagram</b>	Opens the selected SIM file and add it as a sub-diagram to the current diagram.
 <b>Add for loop</b>	Add an empty loop structure to the current diagram.
 <b>Add indexing loop</b>	Add an empty auto-indexing loop structure to the current diagram.
 <b>Properties</b>	Changes the properties of the current diagram.

### The View toolbar

icon	Command
 <b>Full view</b>	Centers the view and manipulates the zoom factor so the current diagram will be fully visible in the diagram view.
 <b>Zoom in</b>	Increases the zoom factor..
 <b>Zoom out</b>	Decreases the zoom factor.

### The Run toolbar

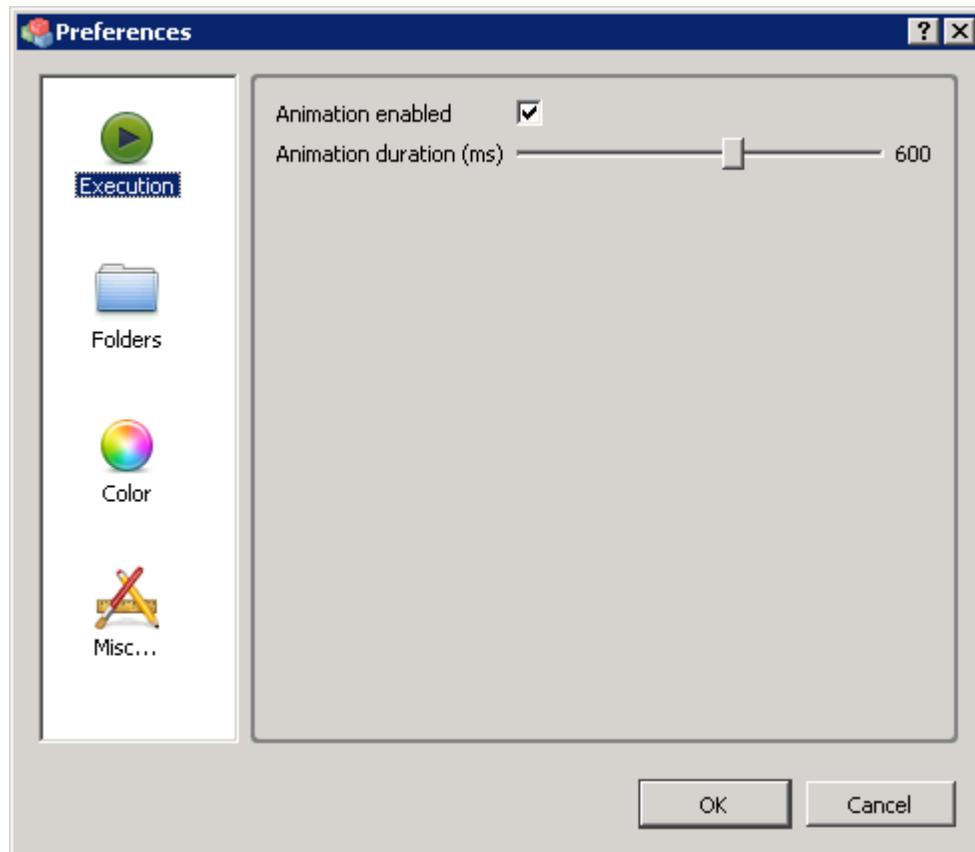
icon	Command
<b>Reset</b>	Resets the generated output values of the diagram.
 <b>Start</b>	Starts the execution of the diagram.
 <b>Step-by-step execution</b>	If the diagram is in design state, starts the step-by-step execution. If the diagram is in paused state, resumes the execution. <i>NB</i> : in both cases, only one step will be executed then the diagram will switch back to paused state.
 <b>Pause</b>	Pauses the execution of the diagram. The currently ongoing execution step will be completed then the diagram will switch to paused state.
 <b>Stop</b>	Stops the execution of the diagram. The currently ongoing execution step will be completed then the diagram will switch to design state.



### 4.14.2.1 Preferences Dialog

This section describes the different parts of the Preferences dialog box.

#### Execution

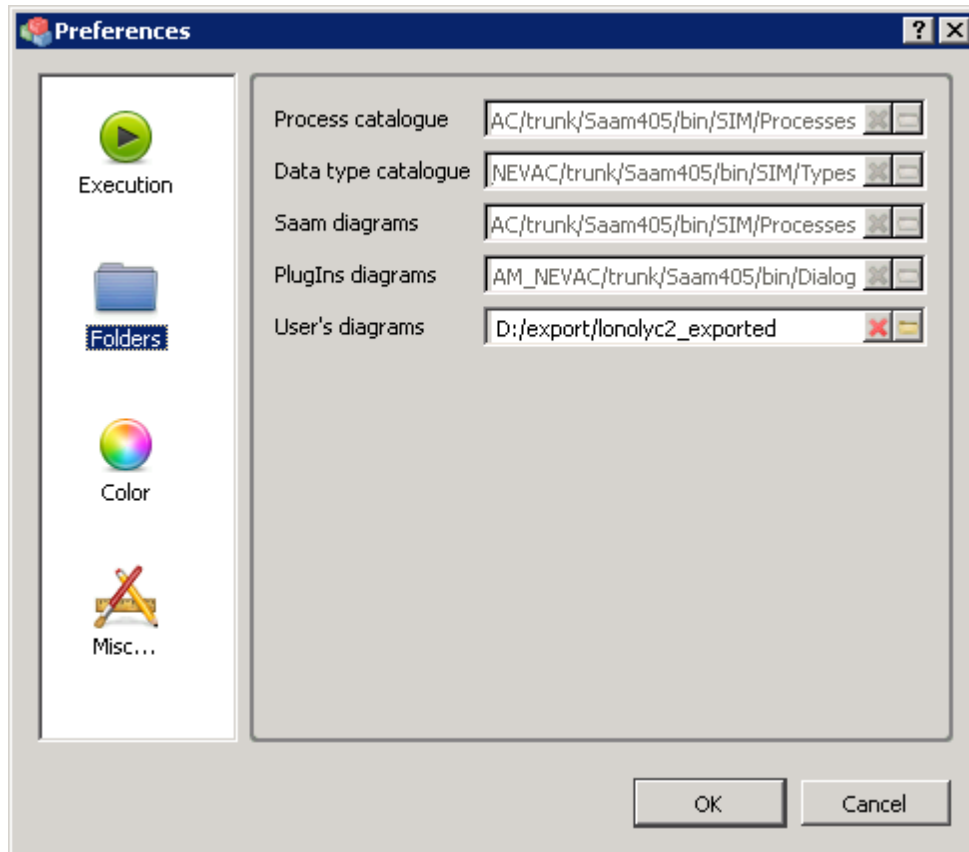


Preferences Dialog : Execution page

The page contains the following items:

1. **Animation enabled** *checkbox* - The tokens are animated along the links
2. **Animation duration (ms)** *slider* - Define the duration of the animation of a token along one link

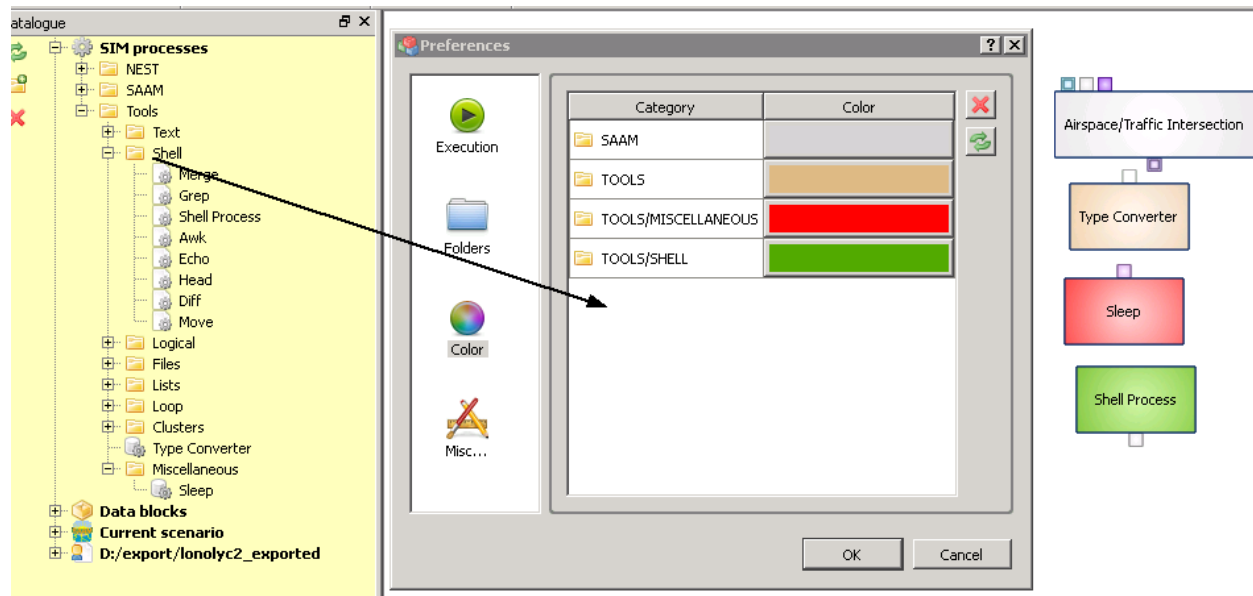
## Folders



**Preferences Dialog : Folders page**

The page lists the active folders to read and load the data types, processes, plugins, predefined diagrams. It's only for viewing purposes to display the working environment of SIM.

## Color



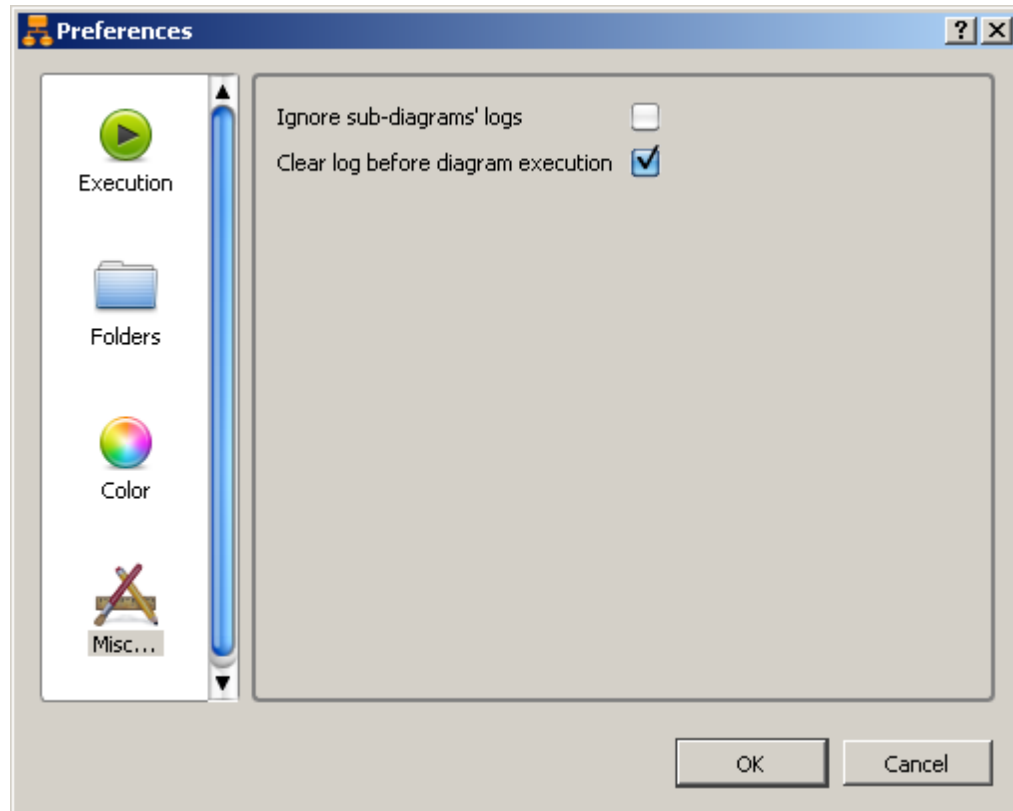
**Preferences Dialog : Colour page**

The page displays a table with two columns defining the colour of the processes. A colour is associated to all the processes that the corresponding "Category", or location in the process tree catalogue, contains. It's useful to group visually by colour, all the processes by type. This colour page is customizable :

- to change an existing colour, click on the colour itself in the "color" column and select a new colour in the Colour dialog popping up,
- to add a new colour, simply drag a folder from process tree catalogue and drop it in the Preferences dialog box. Then, choose one colour,
- to remove an existing colour, click on the associated Category to select it, and then click the red cross to remove it.

Note: the colour associated to a folder defines the colour of all the processes located in any of its sub-folders until a new colour definition is found.

## Miscellaneous



Preferences Dialog : Execution page

The page contains the following items:

**Ignore sub-diagrams' logs** *checkbox* - The log panel contains only top-level (appearing on the graphical scene) diagram execution logged messages.

**Clear log before diagram execution** *checkbox* - The log panel is cleared before each diagram execution.

### 4.14.3 Diagram components

There are five types of components that can be used in a SIM diagram:

- [Terminals](#)
- [Data blocks](#)
- [Processing blocks](#)
- [Loops](#)
- [Sub-diagrams](#)
- [Links](#)





### 4.14.3.1 Terminals

The terminals are the little squares displayed over or/and above the other diagram elements. They are used to attach a link to a block.



**Examples of different terminals**

Different types of terminal:

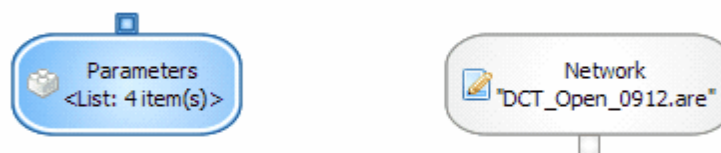
-  A standard terminal. The colour defines the type. It can be linked to another standard or list terminal.
-  A list terminal. The colour defines the type. It can be linked to another standard or list terminal.
-  A cluster terminal. The colour is always yellow. It can be linked to another cluster terminal.
-  An undefined terminal. Can be linked to anything. It takes the type of the terminal it is linked to.

### 4.14.3.2 Data blocks

Data blocks are the terminal points of a diagram and represents its input and output. Each data block has a textual identifier, a direction (input or output), a data type and cardinality. It can also contain a value, which can have a different meaning depending on its direction, type and cardinality.

### Representation on diagram

The following figure shows two examples of SIM data blocks, illustrating the formalism used for their representation on a diagram:



Two examples of data blocks

The element on the left side is an output data block of the string type, with list cardinality, named Parameters, and currently containing 4 values. The element on the right side is an input data block of the ARE file type, with single cardinality, named Network, and currently pointing to the file DCT\_Open\_0912.are.

The following formalism is used to represent data blocks:

- The background colour is the one associated to the data type of the data block.
- The icon on the left is the one associated to the category of data type of the data block.
- The [terminal](#), is displayed below or above the data block, depending on its direction (below for input data blocks, above for output data blocks).
- The name of the data block is displayed on the first text line.
- The current value stored in the data block is presented in the second text line.
  - For file-type data blocks, only the file name is displayed, although the complete path is stored internally
  - For list-cardinality data blocks, only the size of the list is displayed, not its contents.

### Data types

An extensive number of data types are available in SIM, organized in 4 categories:

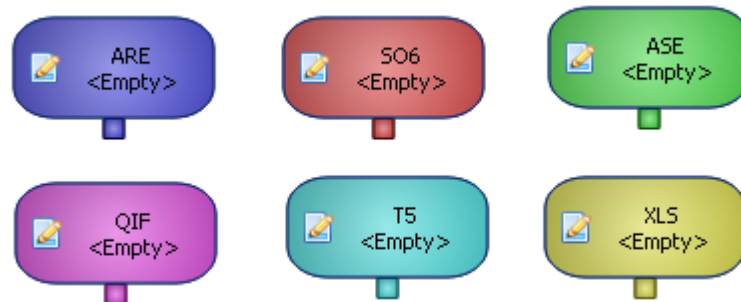
- **Primary types:** This category contains primitive data types as one can find in mainstream programming languages, such as integer, real, string, bool, etc.
- **File types:** This category contains the various file formats available in SAAM, such as SO6, ARE, SLS, etc. Each file type is defined by an extension filter, which will be used in the file browser dialogs displayed by SIM.
- **Enumerative types:** An enumerative type is a list of labels. Enumerative types are typically useful to store program options.
- **Cluster types:** A cluster type is a structured collection of identified elements of any SIM type, even other clusters. The value of the cluster is composed by the values of each of its elements. Cluster types are useful to store a complex set of data having a logic link, for

example an Airspace Network (composed by an ARE file, a SLS file, etc).

## Colours

Each kind of data type is represented by a specific colour in the diagram according to the following rules :

6 different colours are used for all of the Data Types:



### Colours for business types

- Blue for airspace related data
- Red for traffic related data
- Green for network related data
- Magenta for SAAM operating files (QIF files, TDV files, ...)
- Cyan for technical data or intermediate files (T5 file, ...)
- Yellow for system files (text files, Excel files, ...)

In addition, the intensity of the colour varies depending on the associated technical type :



### Colour intensities for technical types

The intensity of the colour is analogous to the amount of information handled by the Data type :

- An Enum type which is a choice in a list is lighter than a primary item,
- A Primary item has the standard intensity level,
- A File that contains a list of items is darker than just one item,
- A Cluster which is a collection of Data types is darker than its components.

## Data cardinality

Two cardinality types are considered in SIM:

- **Single elements:** can only contain one value.
- **List values:** can contain several values of the same data type (not supported for cluster types).

## Optional data block

A data block can be optional. It means that SIM is not expecting the data block to have a real value to start or continue the execution of the diagram.

## Execution visibility

A data block which doesn't have the execution visibility flag on will not appear in the execution preparation dialog box.

## Advanced concepts

### File redirections

As a general rule, the purpose of input data blocks is to allow the user to specify input data for the diagram, regardless of the data type of the block.

However, the purpose of output data blocks depends of the data type. If the data type category is *primary* or *enumerative*, the data block displays the value it receives from the diagram (think of it as an indicator).

But, if the data is a *file*, the data block can allow the user to specify information: the path where the temporary data file received from the diagram will be actually saved. That information is called *file redirection*.

### 4.14.3.3 Processing blocks

Processing blocks represent the individual operations performed and combined by a SIM diagram. Each processing block accepts a given number of inputs, executes a process on it, and then delivers a given number of outputs.

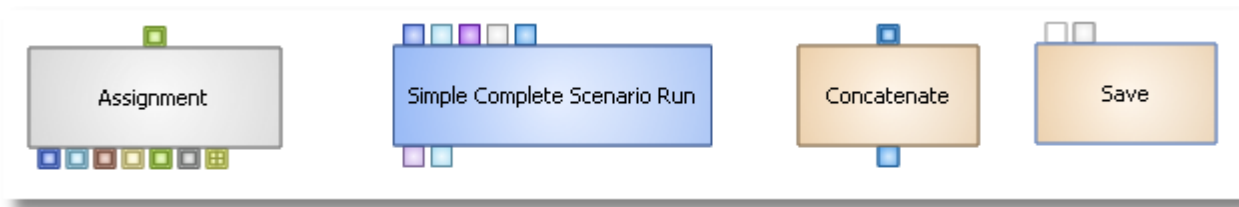
### Representation on diagram

The inputs and outputs of a processing block are called **data slots**. Each data slot is characterized by a name and a data type and cardinality, and is represented graphically by a [terminal](#) located around the processing block. These terminals use the same graphical formalism as the [data blocks](#) terminals.

Input terminals of the processing block are gathered in a line on top of the block. Output terminals are gathered at the bottom of the block.

The name of the processing block is displayed on the block itself. Each processing block has a defined colour, which can be different from one block to the other. Today there is no formalism which define these colours, but in most of the case, the grey processes are the most important ones.





Four examples of SIM processing blocks

#### 4.14.3.4 Links

SIM links are used to establish connections between elements of a SIM diagram.

A SIM link can only be established between an **emitting** terminal (belonging to whichever block in the diagram) and a **receiver** terminal.

Emitting terminals include:

- The terminal of an input data block,
- The output data slots of a processing block,
- The emitting terminals of a loop tunnel

Receiver terminals include:

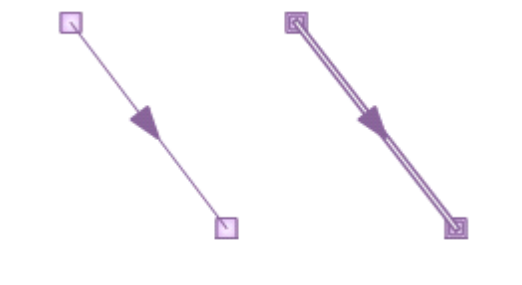
- The terminal of an output data block,
- The input data slots of a processing block,
- The receiver terminals of a loop tunnel

Type consistency must also be ensured when linking slots. Only output slots and input slots of the same type can be linked; the link then takes the same type as the two connectors it is joining.

However one exceptions exists to that type consistency rule. The "*filepath*" type data can be linked to any other data, as long as it belongs to the *file* type category.

A link is represented graphically by a straight line joining the emitting and receiver terminals. It is coloured accordingly to the data types of the terminals it is linking, and has also an optional caption.

The link is drawn using a simple line when it has single cardinality, a double line if it has list cardinality, as shown in the following figure.



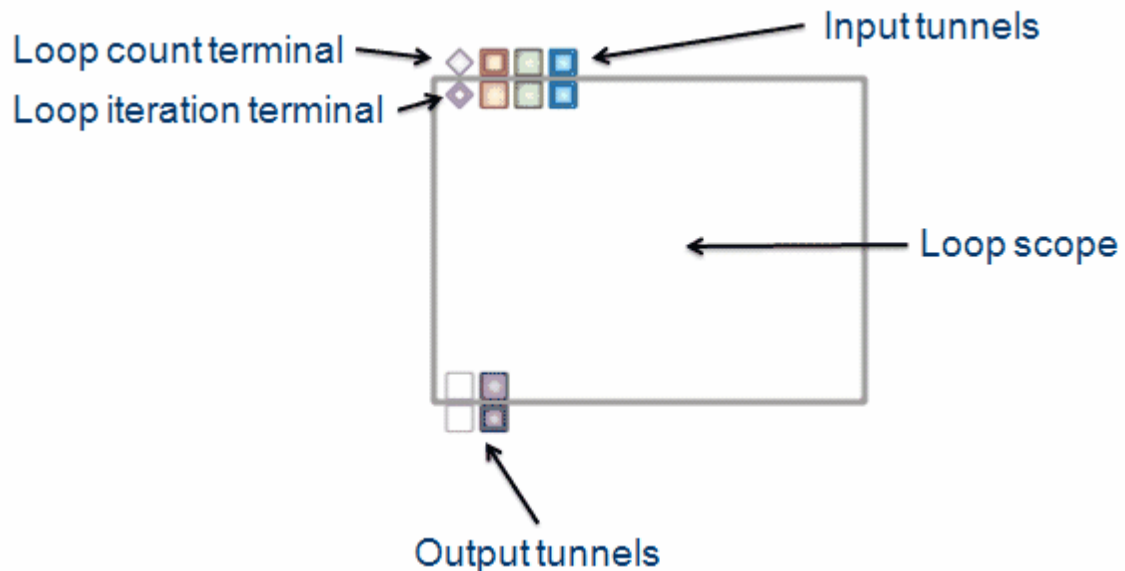
Single-cardinality and list-cardinality link representations

An arrow is drawn to show the direction in which the data is flowing (i.e. from the emitting end to the receiver end).

An emitting terminal can be linked to several receiver terminals simultaneously. However, a receiver terminal is always connected to only one emitting terminal.

#### 4.14.3.5 Loops

SIM loops allow repetitive execution of parts of a SIM diagram. Its formalism is presented in the following figure:



**SIM loop example**

The boundary of a loop is represented by a grey square delimiting the loop scope. The loop scope will contain the diagram that must be executed iteratively.

Special [terminals](#) are part of a loop: the tunnel terminals (paired), the loop iteration terminal and the loop count terminal. The behaviour of these elements is detailed below.

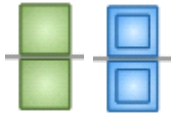
#### Loop tunnels

Loop tunnels are a pair of terminals (one emitting on top, one receiver on the bottom) associated to a loop. They can be placed on the top border of the loop (they are then called input tunnels) or on the bottom border (output tunnels). They always have one terminal on the outside of the loop and one on the inside. Their purpose is to allow data tokens to flow between the loop scope and the main diagram which the loop belongs to.

Both terminals of a tunnel must have the same data type, but not always the same cardinality. This implies several possible configurations, called *tunnel modes*.

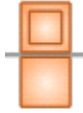
The communication between the emitting terminal and the receiver terminal of a tunnel is called *propagation*. Propagation modalities depend on the tunnel mode.

All available tunnel modes and corresponding modalities are detailed below (tunnel colors are given as example and are not impacted by the tunnel mode):



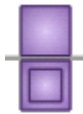
- **Copy 1 → 1 or Copy N → N**

- Mode available to input and output tunnels
- The emitting terminal always propagates the value (single or list) received by the receiver terminal



- **Indexing N → 1**

- Mode available to input tunnels only
- Receiver slot (outside loop) receives a list
- At each iteration of the loop, the emitting terminal emits a new single value coming from the received list (in the same order)



- **Indexing 1 → N**

- Mode available to output tunnels only
- Receiver slot (inside) receives a single value at each iteration
- After last iteration, the emitting slot emits a list containing all the cumulated values



- **Cumulative loop tunnels N → N**

- Mode available to output tunnels only
- Receives a list (of any data type) at each iteration end
- At loop end, emits a concatenated list containing all values of the lists received during execution
- Useful when using nested loops

## Loop types

There exist two types of SIM loops:

- **For loops:** The total number of iterations to execute is passed as an integer value to the loop count terminal. The loop stops its execution after having performed the desired number of iterations.
- **Auto-indexing loops:** These loops have no loop count terminal. Instead, the loop iterates as long as there are values available on its input tunnels, particularly on the indexing ones. The loop stops as soon as one its input indexing tunnels has propagated all the values of the list it received from the diagram in its receiver terminal.

## Integration in execution model

See the [execution model](#) page to learn how SIM run a diagram.

From the outside, a loop is seen just as another processing block: it has some input and output terminals and encloses a processing logic.

Therefore, when a loop is contained in a diagram, it will start its execution just as another processing block: when all its input terminals are ready. The input terminals of a loop are:

- Its loop count terminal (if it has one),
- All receiver terminals of its input tunnels

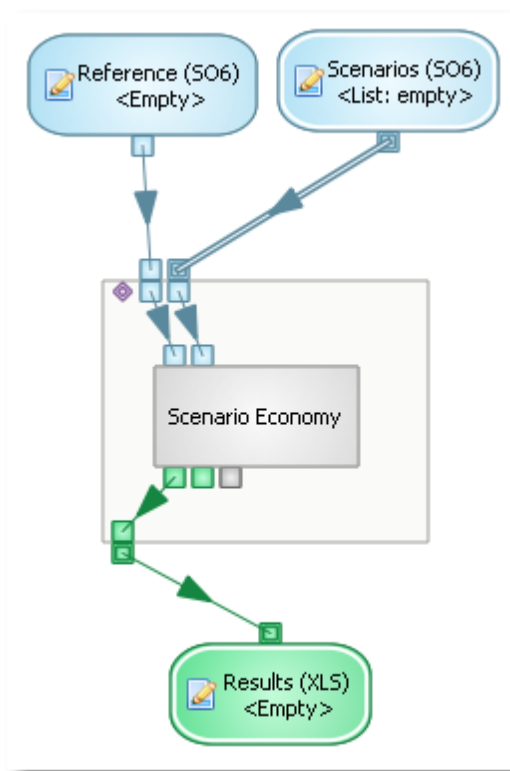
When a loop has finished its processing (i.e. when the last iteration has been completed), values are propagated from the output terminals of the loop to the rest of the diagram.

The execution of one loop iteration is performed by following that scheme:

1. For all input tunnels: propagate data from the receiver to the emitting terminal. The propagation method depends on the mode chosen for each tunnel: copy (the value is simply copied from one terminal to the other) or indexing (the next value from the list on the receiver is extracted and stored on the emitter).
2. Execute the diagram contained in the loop scope. Entry points of that diagram are the loop iteration terminal (containing the index of the current iteration) and the emitting terminals of the input tunnels (which have been filled with values at the previous step). The execution should finish by putting values in the receiver terminals of the output tunnels of the loop (considered as exit points of the diagram).
3. For all output tunnels: propagate data from the receiver to the emitting terminal. The propagation method depends on the mode chosen for each tunnel: copy, indexing or cumulative.

The execution of the loop itself consists simply in executing iterations until the stop condition (depending on the loop type: *for* or *auto-indexing*) becomes true.

### Example



**A diagram with an auto-indexing loop**

The goal of this diagram is to run the scenario economy analysis between a single reference and a list of scenarios.

To do that, we connect the input data blocks to input tunnels of an auto-indexing loop. The reference data block is connected using a copy 1-1 tunnel and the scenarios data block is connected using a cumulative tunnel. During the execution, the loop will automatically iterate over all the file in the scenario list. At each iteration, the scenario economy process is run with the reference and one of the scenario.

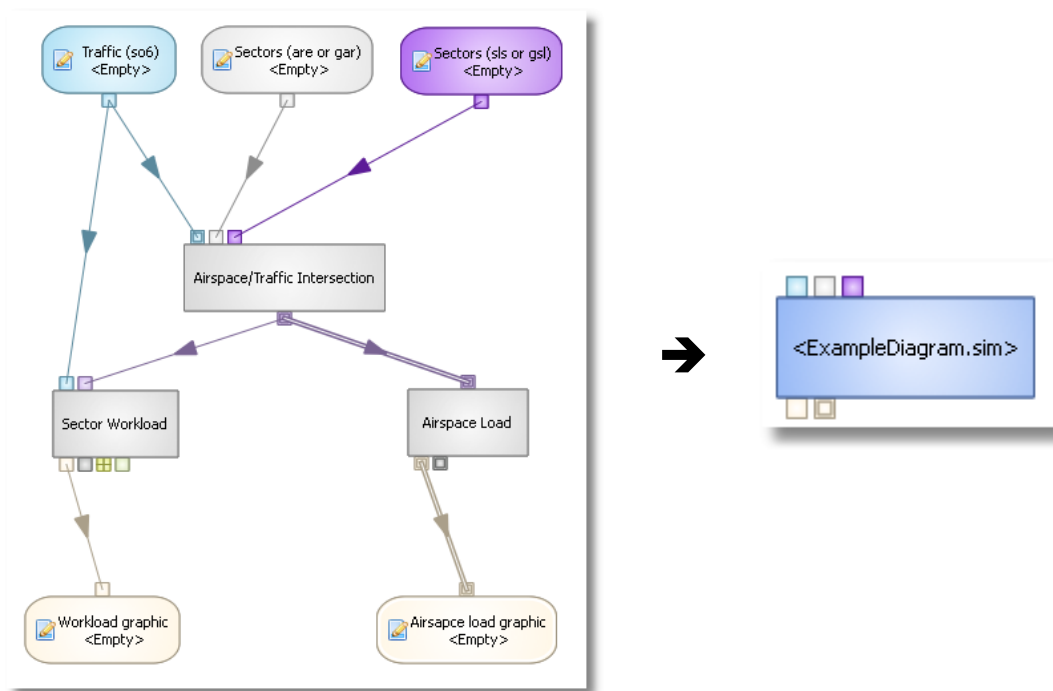
*Note:* If the second input tunnel of the loop was a copy tunnel 1-1, at each iteration the process would have run with the reference and the first scenario.

What we want as output of the diagram is a list containing all the excel result files of the analysis, in the corresponding order. To do that, we connect the output slot of the processing block to an indexing output tunnel of the loop. That way, every value arriving in the tunnel at the end of an iteration will be appended in a list, which will be propagated to the diagram's output data block when the loop will have completed its execution.

#### 4.14.3.6 Sub-diagram

SIM allows a modular diagram design by being able to consider diagrams as ordinary processing blocks, and thus including them as sub-diagrams in any other diagram.

Each data block of the diagram is considered as a data slot of the processing block representing the diagram, as shown in the following figure.



**Example diagram, expanded and collapsed**

The three input data blocks become three input data slots (keeping the same name and type), and the two output data blocks becomes two output data slots.

The sub-diagram then behaves just like any other [processing block](#): it can be connected to other diagram elements.

## Creation of your own sub-diagram

To use one of your diagram as a sub-diagrams, simply save it in your working directory. It will then appear in the [catalog](#). From here, you can drag it on any diagram you want and use it like a normal process.

In order to customize the appearance of your sub-diagram, don't forget to the set its [properties](#) (Name, author, description) in the properties dialog box.

### 4.14.4 Building diagrams

The present section describes the various actions available to the user when building diagrams using SIM Designer.

#### 4.14.4.1 Adding data blocks

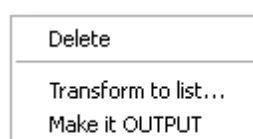
You can add new data blocks to the current diagram simply by drag & drop from the *Catalog* panel: choose one data type in the catalog (organized by category) and drag it to the diagram view. A new data block of the desired data type will be created at that position in the diagram.

Any data block can be moved on the diagram by clicking on it and dragging it to the desired position.

To delete a data block, you can select it and press the Delete key. An action is also available in the menu and toolbars. Either method you choose, SIM will ask you to confirm the deletion of the selected objects.

### Editing data block properties

Once it has been created, the cardinality and the direction of the data block can be changed via the contextual menu.



### The contextual menu of a data block

But you can edit most of its properties via the object properties panel. When a data block is selected, the panel presents a form similar to what is shown in figure below:

Type	Datablock
Name	X Factor
Direction	INPUT
Execution visibility	TRUE
Value	14.04

### Data block properties page

This page allows editing the properties of the selected data block:

- **Name** of the data block
- **Direction** (*input* or *output*)
- **Execution visibility** (*true/false*): defines if the data block will appear in the [execution preparation dialog box](#). more information.
- **Value** of the data block: the nature of this control depends on the type and cardinality of the data block. See next section for more details.

## Setting the data block value

Each data type available in SIM is associated to a particular set of admissible values, and therefore to a particular scheme of user controls to specify these values:

- **Integer** and **real** values are specified using a “spin box” control:

Value	14.04
-------	-------

- **Text string** values are specified using a “line edit” control:

Value	SAAM Intuitive Maker
-------	----------------------

- **Directory** and all **file-type** values are specified using a “file browser” control. The red cross clears the selection, the pen launches the associated file editor, the folder allows to browse for the file:

Value	D:/export/test.qif
-------	--------------------

- **Boolean** values are specified using a “combo box” control allowing only true or false values:

Value	FALSE
-------	-------

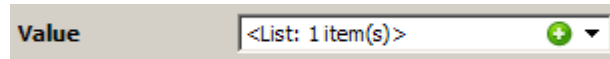
- **Enum-type** values are specified using a “combo box” control allowing only literals belonging to the enum type:

Value	Segment
-------	---------

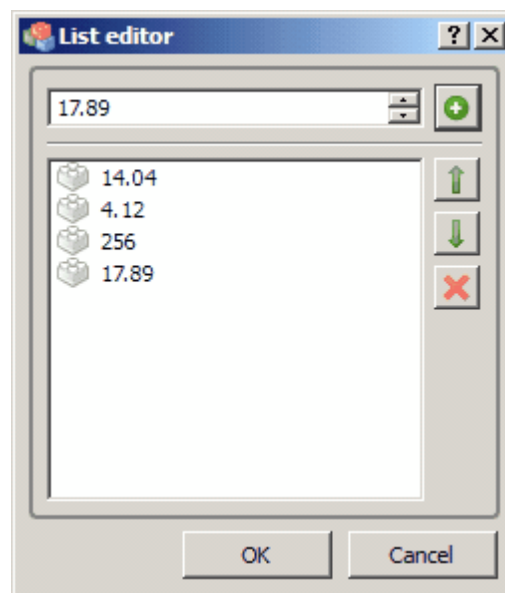
- **Cluster-type** values can't be specified right now. A workaround is to use the process "Cluster Converter" available in the section *SIM Processes/Tools/Cluster* of the [catalog](#).

The behavior of the Object properties panel depends also on the cardinality of the data block:

- For **single-cardinality** data blocks, the Object properties panel simply presents the control appropriate for the type of the data block, ready for you to input values
- For **list-cardinality** data blocks, the Object properties panel presents a list editor control, similar to this one:



This control allows visualizing the current contents of the list, by clicking on the arrow-shaped button at the far right. The contents of the list will then be displayed in a drop-down box. You can also edit the contents of the list by clicking on the “+” button. This will display the List editor dialog box, shown the following screenshot:



**List editor dialog box**

This dialog presents the current contents of the list and allows:

- Reorganizing its order, by clicking on the arrow-shaped buttons on the right
- Deleting elements, by selecting one of them and clicking on the X-shaped button
- Adding a new element to the list, by specifying a value in the control at the top of the dialog box, and clicking on the “+” button on the upper right corner. The type of control displayed depends on the data type of the block.

#### 4.14.4.2 Adding processing blocks

Similarly to data blocks, you can add new processing blocks to the current diagram simply by drag & drop from the [Catalog](#) panel: choose one processing block type in the catalog and drag it to the diagram view.

To delete a processing block, you can select it and press the *Delete* key. An action is also available in the menu and toolbars. Either method you choose, SIM will ask you to confirm the deletion of the selected objects.

You can, if you want, easily create new data blocks on the diagram associated to the data slots of a processing block. Right-click on the processing block and select the “*Add slots...*” submenu; you will see the list of data slots of the processing block. When you click on any of these elements, a new data block will be created on the diagram, having the same



characteristics as the selected data slot (name, type, cardinality), and it will be automatically linked to the selected data slot in the processing block. The “All” item in the menu allows to perform the same task for all slots of the processing block at the same time. Please note that this feature is only proposed as a facilitator and it is possible to obtain the same result by manually creating the data block and the link manually.

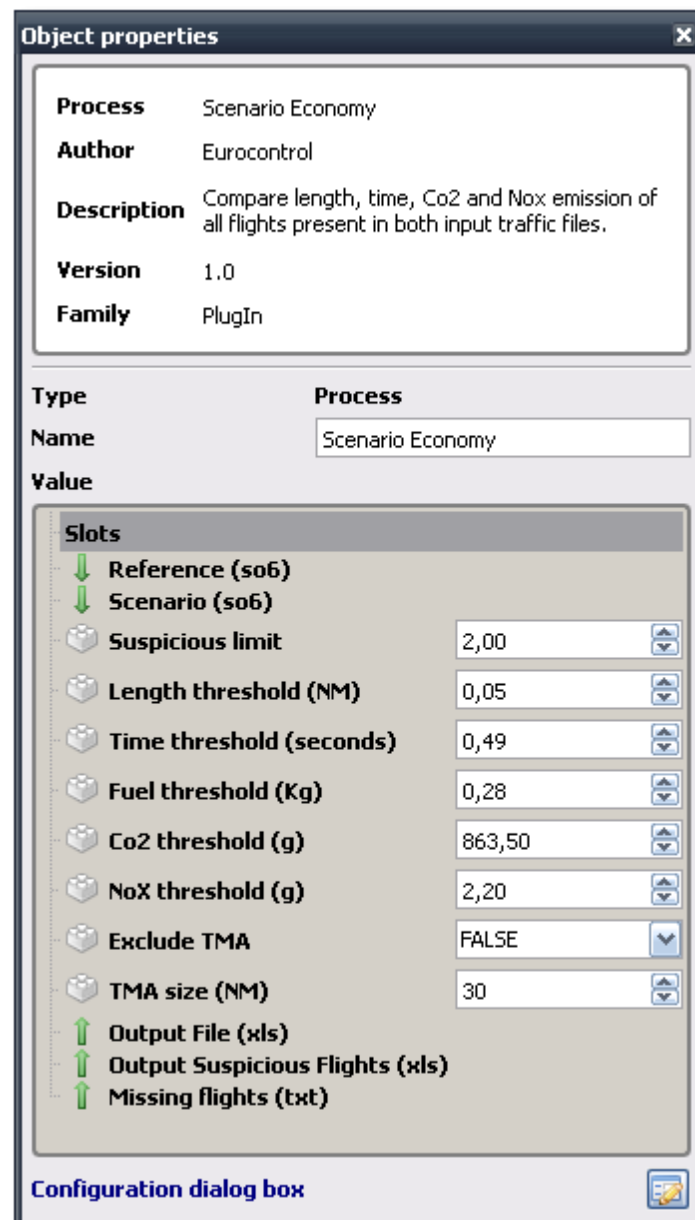


**The context menu of a processing block**

The "*Configure...*" action if the process have a separate configuration dialog box allowing to define values for non-linkable parameters. This is the case for the main SAAM processes.

## Editing processing block properties

Once the processing block has been created, you can edit its properties via the Object properties panel. When a data block is selected, the panel presents a form similar to what is shown in Figure 16:



The properties of the "Scenario Economy" process

This page presents the data of the processing block (in the white frame), its name, and its data slots presented in a tree-like structure.

You can switch the data slots of the processing block between **linkable** and **non-linkable** modes by right-clicking on the name of the slot in the tree and selecting "Set to linkable" (or "Set to non-linkable, respectively). The two modes differ by the way the values are obtained:

- In **linkable** mode, a terminal is associated to the data slot in the processing block's representation on the diagram view. The slot will then receive values from other elements of the diagram, at execution time.
- In **non-linkable** mode, no terminal is associated to the slot. Instead, the diagram designer is prompted to set a constant value that will not change during the diagram execution. The value can be specified directly in the Processing block properties page, in the tree-like slot list.

Some processes may also have a separate configuration dialog box allowing to define values for non-linkable parameters. In that case, you will see a specific button at the bottom of the properties page:

Configuration dialog box



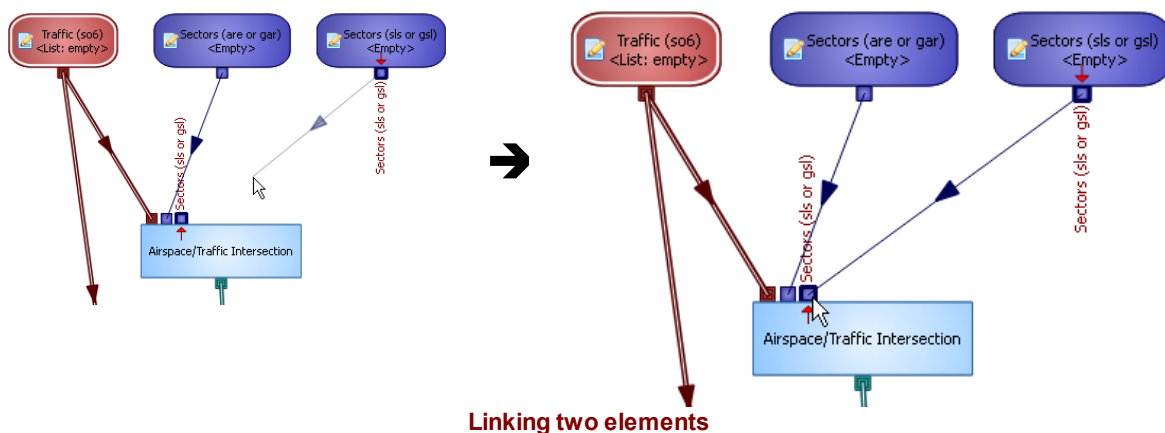
Clicking this button will display the dialog box, whose appearance is specific to each process. This option is also available by the "Configure..." action in the context menu of these processes.

A configuration dialog box is available for the main SAAM processes. It is exactly the same dialogs that you can find in the SAAM menus, except that the input data which are linked are not available.

#### 4.14.4.3 Linking elements

##### Creating a link

To create a link between two elements of the diagram, you simply have to drag the mouse cursor from one terminal to the other. SIM Designer will highlight any compliant terminal with a red arrow under it, and display the associated caption for further discrimination in case of several compliant terminals in the vicinity. SIM designer will also confirm that the connection is possible by automatically "snapping" the link extremity to the terminal. You can drag the cursor from the emitter to the receiver terminal or inversely, the flow direction will be corrected if needed.



You can add a caption to an existing link by double-clicking on it and typing the caption in the dedicated dialog box. A link can be deleted similarly to other diagram elements, by selecting it and pressing the Delete key.

##### Links and cardinality

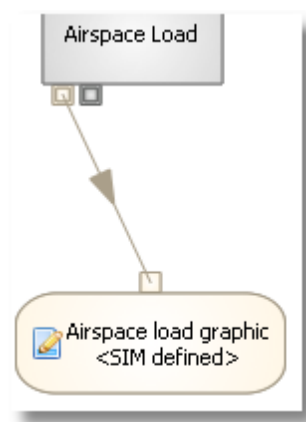
In most of the case, it is advised to always link to terminal with the same cardinality. However, it is possible to link two terminal of different cardinality.

When an emitting single terminal is linked to a receiver list terminal, the link will transform the single element in a list containing one element.



A link between an emitting single terminal and a receiver list terminal

When an emitting list terminal is linked to a receiver single terminal, only the first element of the list is transmitted through the link.



A link between an emitting list terminal and a receiver single terminal

#### 4.14.4.4 Adding loops

[Loops](#) can be added to the diagram by clicking on the corresponding buttons on the toolbars or in the menu (See the [menus and functions](#)), corresponding respectively to for loops and auto-indexing loops. Clicking on one of these buttons will create a new, empty loop structure on the diagram.

### Adding contents

To add contents (processing blocks or other loops, data blocks are not allowed) into the scope of a loop, select the elements to be added and drop them into the loop scope **while holding the Ctrl key**. The border of the loop are highlighted in red if you can drop your element inside. To remove an element from a loop, hold the SHIFT button.

Elements in a loop scope can be moved with the mouse but are bounded to the loop scope. To take elements out of a loop and put them back on top level of the diagram, hold the *Shift* key while dragging these elements outside of the loop.

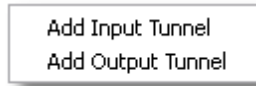
Linking elements in a loop scope is done the same way as outside of the loop: simply drag a link between the two terminals to connect. It is also worth noting that cross-scope linking is not allowed. Instead, SIM will automatically create and connect the necessary tunnels to permit that connection.

To delete a loop, you can select it and press the *Delete* key. SIM will ask you to confirm the

deletion of the selected objects. All elements contained in the loop will be deleted as well.

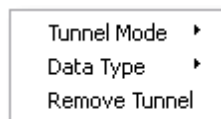
## Adding tunnels

New tunnels can be added to a loop by right-clicking anywhere on the loop scope and choosing either “Add input tunnel” or “Add output tunnel”.



**The context menu of a loop**

Once it has been created, a tunnel must be configured through its contextual menu or directly by linking it.

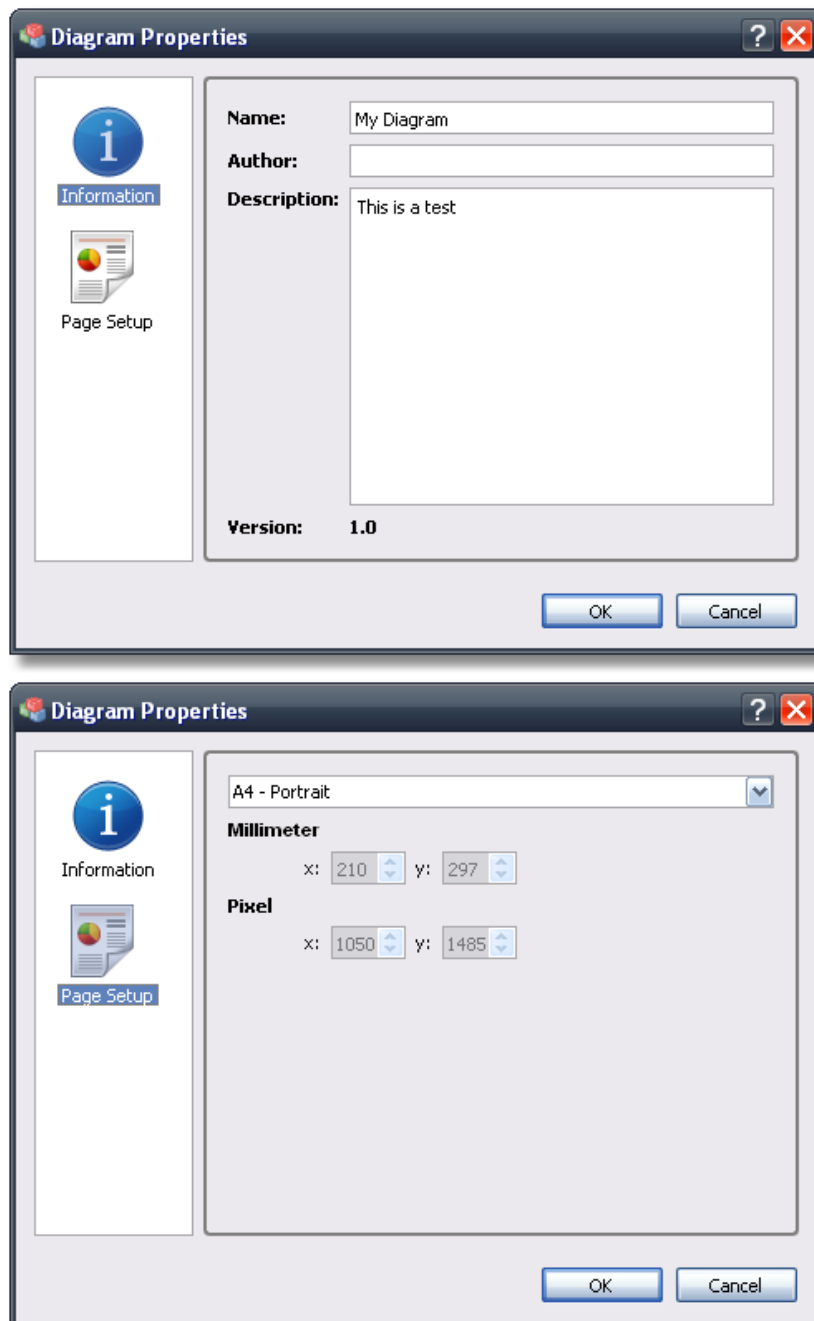


**The context menu of a tunnel**

Note: The tunnel mode can only be configured after than the data type has been selected.

#### 4.14.4.5 Editing diagram properties

The Diagram Properties dialog box contains two pages. The first one (*Information*) allows editing the metadata of the diagram (name, author, comment). The second one (*Page setup*) allows editing the size (in paper dimensions) of the graphical canvas on which the diagram is drawn.



The diagram properties dialog

#### 4.14.4.6 Building output file names

In most of the cases, to retrieve the output of the diagram, the user can simply defined the file redirection parameter of the output [data block](#) where he want the file to be copied after the diagram completion. But sometimes, it is useful to generates the names of the output files from the names of the inputs to speed up and simplify the filling of the execution preparation dialog.

It is possible to do this with the *Easy File Saver* Process block.

Below is an example of diagram using it :

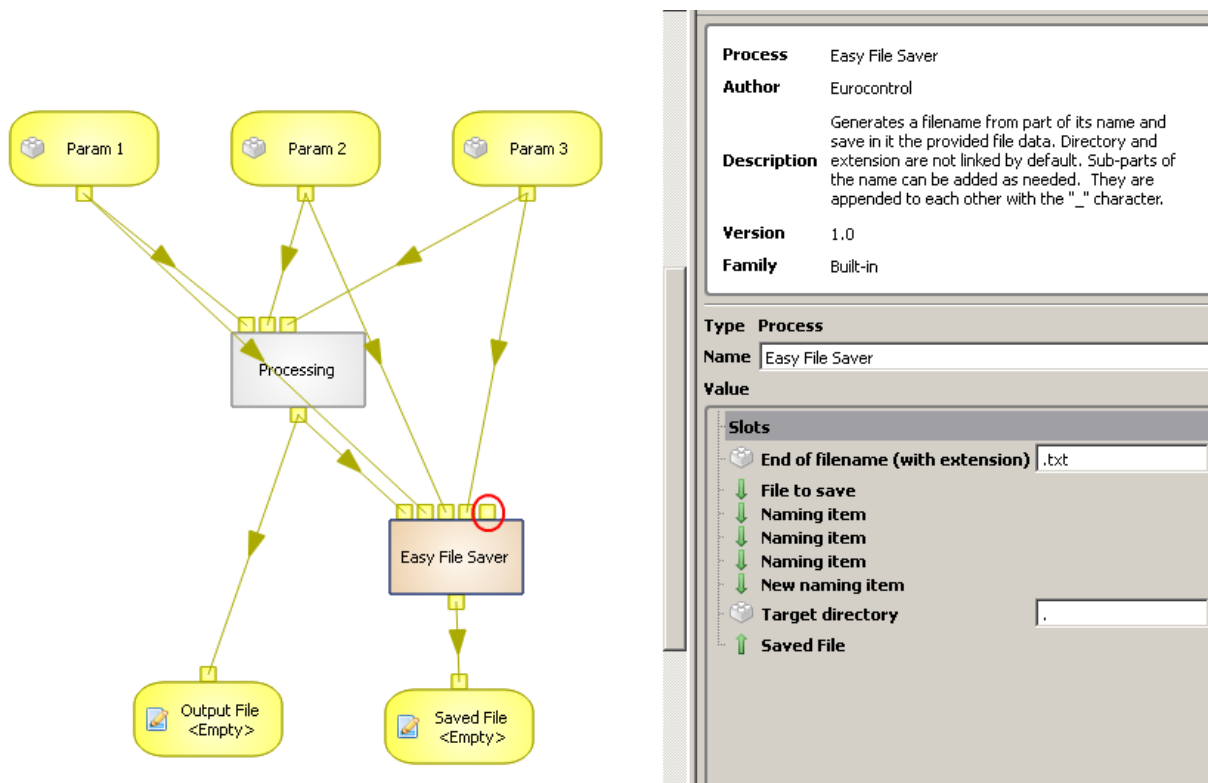


Diagram using the *Easy File Saver* Process block

Let's consider a "Processing" Process block outputting a file according to 3 input parameters that we want to be reflected in some way in the output file. 5 types of information can be connected to the "Easy File Saver" Process block :

- "File to save" : this is the data stream to be connected from the output of the "Processing" Process block,
- "Saved File" : this is the input data stream saved into the desired output file. Generally, it's not useful to connect it,
- "Naming Item" : this is a String specifying one piece of the output file name. There may be as many pieces as needed, each of them being automatically concatenated with the "\_" character to build the final output file name. Thus, in the example diagram, it would be Param 1\_Param 2\_Param 3. To add a new naming item, simply connect another String dataslot to the "New naming item" dataslot (highlighted with a red circle on the picture). It would automatically generate a new input data slot,
- "Target directory" : specifies the path of the filename to be saved. By default, it's the current directory (".") but it's possible to make it linkable to change it,
- "End of filename" : specifies a suffix to the filename. It may be a convenient way to specify the file extension, but is not restricted to the extension. It can be empty.

## Additional changes on the output data block

As the output file is now written by the *Easy File Saver* process block, the output data block is not really necessary anymore except for the visualisation of the diagram and if you want to use it as a [sub-diagram](#).

If you keep it, you can set its "Execution visibility" property to false and leave its "File redirection" property blank. So it won't appear in the execution preparation dialog. Mind then, that it wouldn't appear either as an output slot if the diagram is used as a sub-diagram !

### 4.14.5 Running diagrams

#### The SIM execution model

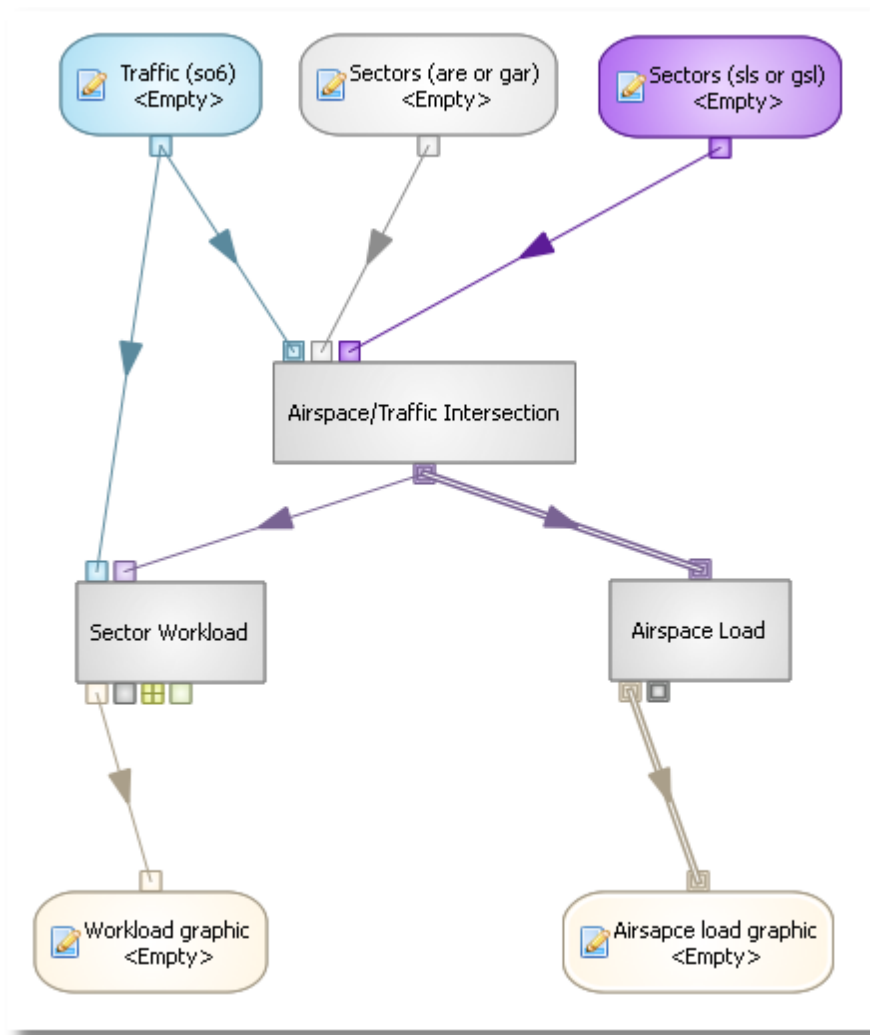
The whole purpose of creating and managing SIM diagrams is to be able to execute them, that is, given a set of input values applied to its input data blocks, flow through the diagram and execute the tasks associated to the processing blocks of the diagram, producing values for the output data blocks of the diagram. The execution of a SIM diagram is defined by a set of rules called the **SIM execution model**.

The SIM execution model defines the order of execution of the processes of the diagram using a token-based approach:

- At the start of execution, all input data blocks release one token (symbolizing the availability of their data) through their emitting terminal.
- Any token delivered through an emitting terminal is transmitted to all receiver terminals that are connected to it.
- A processing block can only execute its underlying process when all its input slots have received a token (i.e. when all inputs are ready).
- When the execution of a processing block is finished, one token is released on all of its emitting terminals.
- The execution of the diagram is completed once no more processing blocks can be triggered, and there is no more tokens circulating in the diagram, besides those that reached output data blocks.

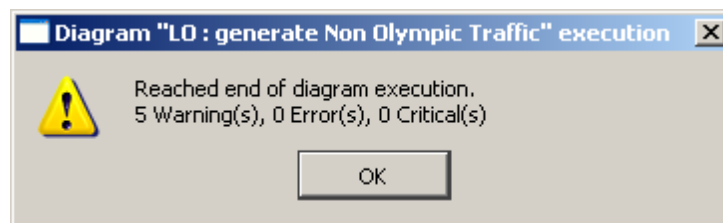


## Execution example




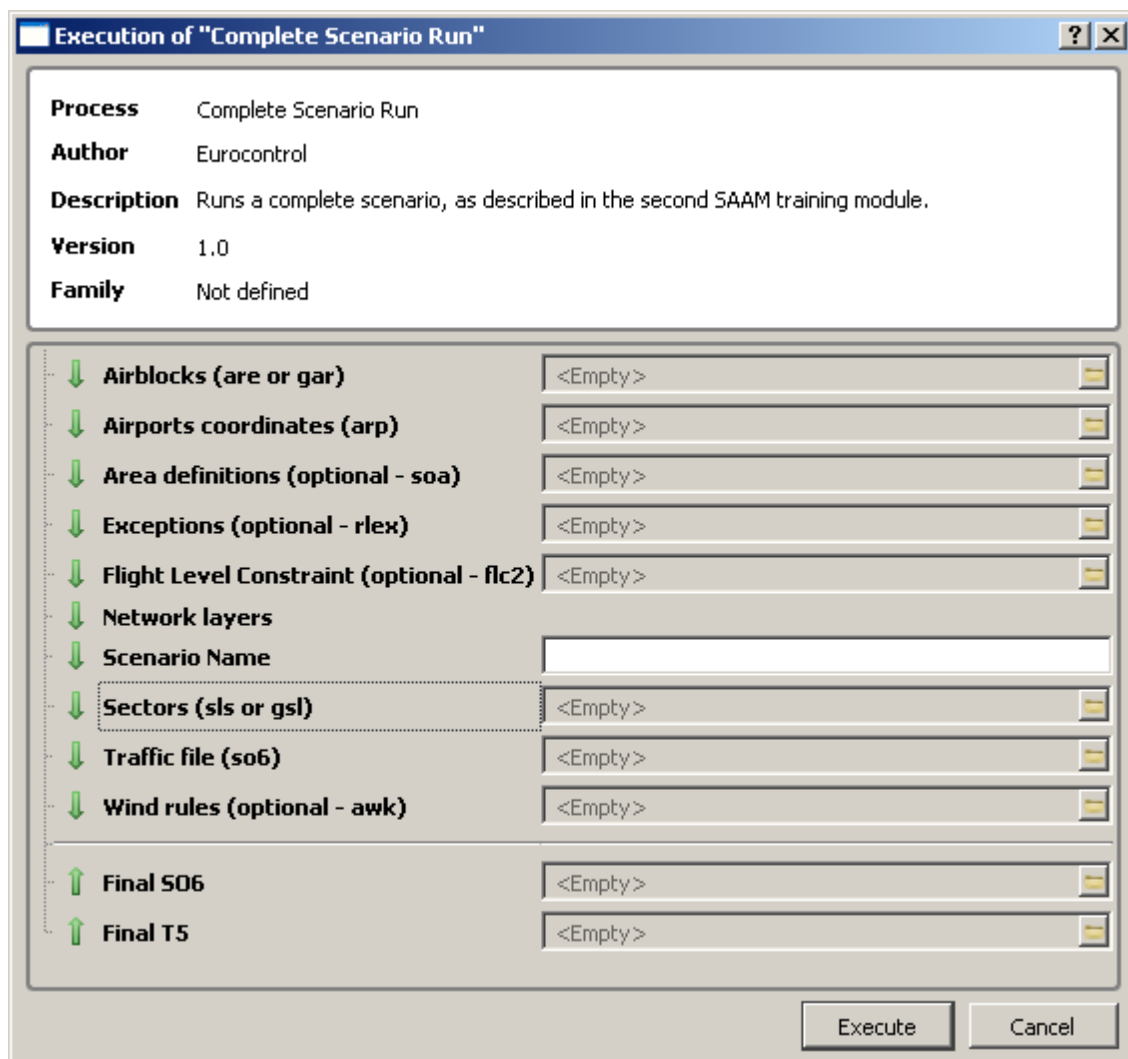
Here is how the diagram above would be executed by SIM:

- Start of execution: all three input blocks release a token on their respective terminals.
- Step #1: Only the "Airspace/Traffic intersection" block has all its input slots ready.
  - Execution of "Airspace/Traffic intersection" block. Its input data is retrieved from its input slots and passed to its underlying process. Upon completion, its lone output slot transmits a token to the "Sector Workload" and "Airspace Load" blocks.
- Step #2: "Sector Workload" and "Airspace Load" block have their inputs available at that moment (from step #1 and from step #2).
  - Both blocks are executed in an unknown order. Output tokens are transmitted through their output slots
- End of execution: There is no block left ready to execute, The two output data blocks have received a token a token and therefore writes the received data on disk at the file redirection location (if specified). Execution of the workflow is completed. A dialog pops up to show the execution diagnostics :



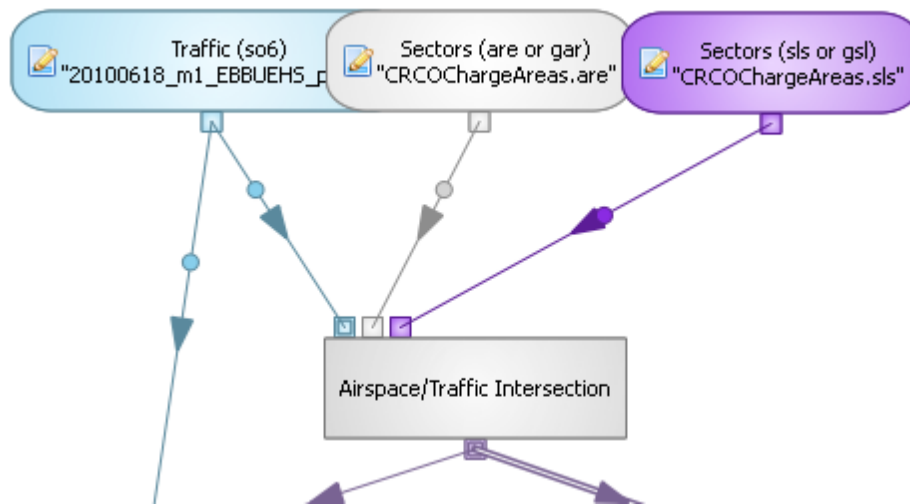
## Running a diagram

Click on the  in the "run" toolbar button to launch the diagram execution. A dialog box will appear, allowing the user to verify/enter all values of the data blocks of the diagram before performing the execution. Input data appear on top of the parameters list, output ones at the bottom:



**The execution preparation dialog box**

The diagram execution is animated: you can see the data tokens flowing through the links from block to block, and the processing block are coloured in green when their execution is in progress. It is possible to completely disable animation in the global SIM preferences (see the [menus and functions](#)).



**A part of a SIM diagram during an animated run**

A diagram execution may be paused by clicking the corresponding button. The pause state becomes effective only after the current processing block has completed its execution, though.

The diagram execution can also be completely stopped. Here again, the stop becomes effective only after the current processing block has completed its execution.

Step-by-step execution is also possible via the “Step over” action available in the menu and toolbars.

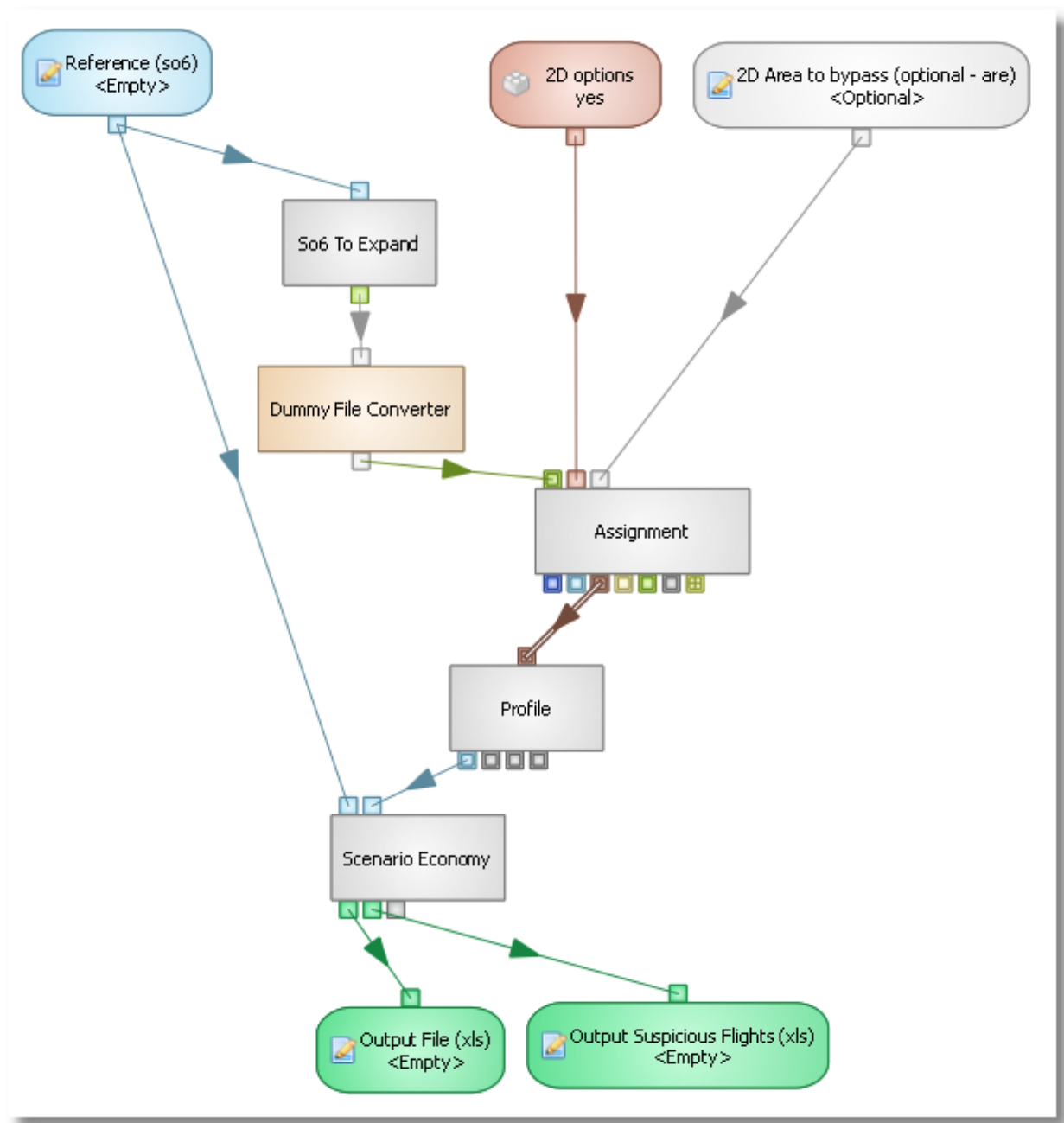
#### 4.14.6 Tutorials

This section presents some tutorials to help learning the basic feature of SIM.

#### 4.14.6.1 Tutorial 1: Scenario economy comparison

The purpose of this tutorial is to build a diagram which will make the scenario economy comparison between a reference traffic file and a scenario traffic calculated by the assignment after that we made a change on our environment.

Here is the final diagram that we want to build in this tutorial:



**Tutorial 1: final diagram**

### Adding the main process blocks

From the catalog panel, drag the processes **Scenario Economy** and drop it on the diagram. You will find it in "SIM processes/SAAM/Analysis".

Add the processes **Profile** and **Assignment** on your diagram using the same method. You will find it in "SIM processes/SAAM/Processing".

Add the processes **SO6 to Expand** on your diagram using the same method. You will find it in "*SIM processes/SAAM/Transform*".

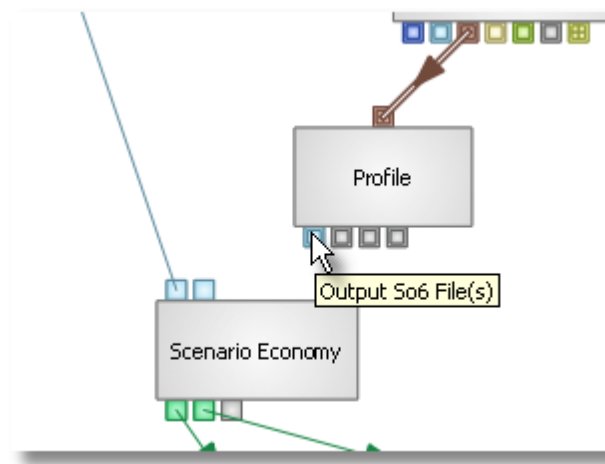
## Adding the main data blocks (inputs/outputs)

On our diagram we want only one main input: a reference traffic. There are two way to add it. You can drag it from the catalog (You will find it in "*Data blocks/File/SO6*") or you can right-click on the scenario economy process block to open its contextual menu and select "*Add slots -> reference (so6)*".

On output, we would like to retrieve the results of scenario economy. Right-click on the scenario economy process block to open its contextual menu and add the output slots that you interested in. For instance, if you do not need the suspicious flight you can simply ignore it.

## Linking the blocks

The next step is to link the blocks together. This is quite easy to do if you know which terminal correspond to which type. To get more information about a [terminal](#), move your mouse over it to display the tool tip.



The tool tip of a terminal

There is one problem. You cannot connect the output slot of the SO6 to Expand process to the corresponding input of the Assignment process even though they have the same type and the same colour. This is due to a current limitation of SIM. In fact, the Assignment is taking as input an EXP **or** an EXP2 and the SO6 to Expand is always generated an EXP file (not EXP2). SIM currently consider this as two different types and prevent the creation of the link between these two terminals.

If you look in the data block catalog, you will find three types for EXP files:

- EXP=Only EXP
- EXP(2)=EXP or EXP2
- EXP2=Only EXP2

Hopefully, this limitation will be removed in a future version of SIM. To solve this problem today, we have to introduce an additional process called **Dummy File Converter** (You will find it in "*SIM processes/Tools/Files*"). This process works as a workaround to link two terminals of different type. Note that the content of the file is not changed and no check is performed to see if the connection you do is meaningful. For instance, you could by this method link the EXP file to a process which requires an SO6 file, but the execution of the

process will probably fail or the results will be wrong.

## Setting the processes parameters

Now the structure of the diagram is completed, but we need to set the internal parameters of each processes. This is particularly true for the assignment which require an high number of parameters. The other processes can be run with their default parameters.

You could configure the processes with the object properties panel but, when available, it is easier to use the configuration dialog. Right-click on the Assignment block and select "Configure..." to open [assignment dialog box](#). You can note that the selection of the input traffic demand and the outputs are disable. This is absolutely normal since they are managed to the assignment by the diagram. However, the other parameters are available. Fill the necessary parameters and save them.

## Adding additional inputs parameters


The diagram could be finished like this. But let's say that you want to have the possibility to run the assignment with a 2D option without changing the diagram. You can do that by making the 2D option parameter of the assignment block "linkable".

Select the Assignment block and, in the object properties panel, right-click on the "2D option" parameter and select "Set to linkable". A new terminal, of type BOOL, now appear on the assignment block. You can now add the input datablock with the contextual menu. Do the same thing for the "2D area to bypass" parameter of the assignment.

Like this, we can add every parameters we want as input of the diagram to be able to manipulate them easily.

One more thing: the "2D Area to bypass" data block should be flag as optional. Indeed, if the 2D options block is set to "no", the 2D area file become unnecessary. By making it optional, SIM won't complain if it is not here.

## Running the diagram

The diagram is now complete. Click on  in the toolbar to open the execution preparation dialog, enter the inputs/outputs files and start the execution.

Watch carefully the log panel to detect the possible warnings and errors.

### 4.15 SIM Extension Guidelines

This document presents the facilities provided by the SAAM Intuitive Maker software to extend its features.

The [first section](#) details how to add new data types to SIM by means of the SDT file and the [second section](#) presents how to add new process blocks ready for use in SIM diagrams, by means of various integration techniques.

### 4.15.1 Data types extensions

Each dataslot in a SIM process or datablock in a SIM diagram has a defined data type taken from the *Data Type Catalog* managed internally by SIM. The major part of that catalog is defined in a XML file (using the SDT format defined for that purpose) read by SIM at start-up. New data types can be added to SIM easily by editing this file.

The next section details the organization of data types within the data type catalog.

### Data types categories

Each data type in SIM has a unique ID and a color (which will be used in SIM designer to represent links, data blocks and data slots having that data type).

Four data type categories are defined in SIM:

- **Primary types:** This category contains primitive data types as one can find in mainstream programming languages. The available primary types are: INTEGER, REAL, STRING, BOOL, DATE, TIME, FILEPATH.
  - The list of primary data types registered in the catalog is explicitly defined in the SIM source code, along with the controls and widgets used to manipulate these data types. It cannot be extended through the SDT file.
- **File types:** Each file type is defined by an extension filter, which will be used in the file browser dialogs displayed by SIM.
  - The extension filter must follow the principles of the [QFileDialog class API](#). The string "SO6 files (\*.so6)" is therefore a valid extension filter.
  - New file data types can be declared through a SDT file.
- **Enumerative types:** An enumerative type is a collection of literals which are implicitly associated to integer values. The user will only see and manipulate the literals, and SIM will manipulate the corresponding integer values internally.
  - Concept similar to C/C++ enums.
  - New enumerative data types can be declared through the SDT file.
- **Cluster types:** A cluster is a structured collection of identified elements of any SIM type, even other clusters. The value of the cluster is composed by the values of each of its elements.
  - Concept similar to C/C++ structs.
  - New cluster data types can be declared through the SDT file.

### SDT file format

A SDT file contains one *Types* element, which has no attribute, no content, and can only contain *FileType*, *EnumType* and *ClusterType* child elements.

### FileType elements

A *FileType* element has the following attributes:

- **name:** Identifier of the type in the catalog and in the toolbox. Must be unique.
- **filter:** File extension filter associated to the type. This will be used for example in file browser boxes to select an input file for a file data block in a diagram.

- **colour:** Colour representing the type. It must conform to the specification of [QColor::setNamedColor\(\)](#). Unicity of the colour among other types in the catalog is preferable but not needed.

Example:

```
<FileType name="So6" filter="SO6 files (*.so6)" color="#87ceeb"/>
```

### EnumType elements

An *EnumType* element has the following attributes:

- **name:** Identifier of the type in the catalog and in the toolbox. Must be unique.
- **colour:** Colour representing the type. It must conform to the specification of [QColor::setNamedColor\(\)](#). Unicity of the colour among other types in the catalog is preferable but not needed.

An *EnumType* element accepts *EnumValue* child elements, one for each literal composing the enumeration (the literal is defined by the attribute value).

Example:

```
<EnumType name="DirectionEnum" color="#669759">
  <EnumValue value="EW"/>
  <EnumValue value="NS"/>
</EnumType>
```

### ClusterType elements

A *ClusterType* element has the following attributes:

- **name:** Identifier of the type in the catalog and in the toolbox. Must be unique.
- **colour:** Colour representing the type. It must conform to the specification of [QColor::setNamedColor\(\)](#). Unicity of the colour among other types in the catalog is preferable but not needed.

A *ClusterType* element accepts *ClusterElement* child elements, one for each element composing the cluster. The identifier and type of each element are defined in the *name*, *category* and *type* attributes.

Example:



```
<ClusterType name="Sectors" color="#dbe659">
  <ClusterElement name="AreFile" category="File" type="Are"/>
  <ClusterElement name="SlsFile" category="File" type="Sls"/>
</ClusterType>
```

## Distribution

To distribute new data types with SIM, there is two possibility:

- You can simply edit the *environment.sdt* file shipped with SIM
- You can add a new SDT file in the SIM data type directory. **Warning:** Be careful about dependencies. The sdt files are read in alphabetic order.

*Note:* the location of the SIM data type directory can be view from SIM preference dialog.

### 4.15.2 Processes extension

The present section illustrates how SIM can be extended with new process blocks ready to be used in any diagram.

Three process integration techniques currently coexist in SIM. These techniques are briefly presented in the first paragraph, then fully explained in subsequent sections. Depending on the characteristics of the process to integrate into SIM, the developer will have to choose the most suited technique.

## Available integration methods

There are currently 4 methods proposed:

- **SPI external processes:** Suited for the integration of standalone, command-line executables or scripts compliant with clearly identified rules. The “interface” of the external tool must be described in a text file using the SPI format, designed for that purpose.
- **Plug-in processes:** Suited for process blocks whose execution code would preferably be developed in C++, but intended to be distributed individually as external DLLs rather than be statically integrated in SIM.
- **Diagrams:** It is worth noting that, since each diagram created using SIM Designer can be integrated as a sub-diagram in another diagram, a SIM diagram is therefore de facto a new process block ready to be used in any diagram. Please refer to the [SIM User Manual](#) for information on how to design diagrams using SIM Designer

#### 4.15.2.1 SPI processes

### General principle

SPI stands for SIM *Process Interface*.

This technique allows to integrate external command-line tools (either .exe binary executables or .ksh shell scripts) into SIM.

It is performed by editing a simple text file (with the extension .spi, compliant with the format described here below) describing the characteristics of the process to integrate, in order for

SIM to know how to present and use the process. These characteristics include:

- Unique ID of the process
- Appearance of the process box on the diagram: colour, caption
- Number, names and types of input parameters
- Number, names and types of output files
- Command-line structure, with placeholders for parameter values

No C++ code (neither from a SIM component or another library) needs to be created nor modified to perform this integration.

## SPI file format

A SPI file contains one Process element, with the structure detailed below.

### Process element

A Process element accepts:

- 8 attributes:
  - **id**: ID (alphanumerical string) of the process, for internal use by SIM. It must be unique throughout the whole pool of processes of SIM.
  - **model**: Must be set to “process” in SPI files.
  - **caption**: Text string that will be displayed as caption of the process block in a SIM diagram.
  - **author**: Text string that will be displayed in the metadata of the process in SIM Designer.
  - **description**: Text string that will be displayed in the metadata of the process in SIM Designer.
  - **version**: Version of the present SPI file.
  - **cmdLine**: Calling syntax of the process.
  - **colour**: Text string describing the colour of the process block in a SIM diagram. The colour specified must comply to the same rules as the [\*QColor::setNamedColor\(\)\*](#) method of the Qt framework, which means that “darkorange” and “#FFA500” are both valid colour strings.
- No content
- 2 types of child elements:
  - **InputParam**: Represents an input slot of the process.
  - **OutputParam**: Represents an input slot of the process.

### InputParam element

An *InputParam* element accepts:

- 7 attributes:
  - **id**: ID (alphanumerical string) of the slot. It must be unique in the present SPI file.
  - **caption**: Text string that will appear in SIM Designer to represent the slot, for example in tooltips.
  - **category**: Text string representing the category of the slot’s data type. The acceptable values are: “PRIMARY”, “FILE” and “ENUM”.
  - **type**: Text string representing the slot’s data type. This must correspond to a registered

[data type](#) in SIM's catalog.

- **cardinality**: Must be “single” for single values, “multi” for lists.
- **linkable**: Boolean (“yes”/“no”) value indicating if the slot is linkable (i.e. it appears on the diagram) by default (the slot configuration of a particular process block on a diagram can be changed at any moment).
- **optional**: Boolean (“yes”/“no”) value indicating if the slot is optional. False by default.
- No content
- No child elements

## OutputParam element

An *OutputParam* element accepts:

- 4 attributes:
  - **id**: ID (alphanumeric string) of the slot. It must be unique in the present SPI file
  - **caption**: Text string that will appear in SIM Designer to represent the slot, for example in tooltips.
  - **category**: Text string representing the category of the slot's data type. For output values of a SPI process, only the FILE category is allowed.
  - **type**: Text string representing the slot's data type. This must correspond to a registered [data type](#) in SIM's catalog.
  - **optional**: Boolean (“yes”/“no”) value indicating if the slot is optional. False by default.
- No content
- No child elements

NB: a SPI process can only have outputs of “single” cardinality and belonging to the “file” category.

## Calling syntax representation

The *cmdLine* attribute of a process is intended as the only way for the developer of the process to “inform” SIM about the command-line interface of the process and the way it should be called.

This information is stored as a generic command-line string, containing special tokens referring to the declared data slots of the process block. For any data slot (named slot for the example), the following tokens can be used in the command line:

- **%slot**
  - Available for single-cardinality slots only
  - Placeholder for the value contained in the data slot
- **(%slot)**
  - Available for multi-cardinality slots only
  - Placeholder for the concatenated values of the list contained in the data slot
  - Values are separated by whitespaces
- **[%slot]**
  - Available for multi-cardinality slots only
  - Placeholder for the size of the list contained in the data slot

At process execution time, SIM will process the generic command line and replace all the tokens occurrences according to the actual values contained in the slots of the process. The resulting command line obtained is passed down to the shell (Cygwin bash) for execution.

## Examples

Here are a few examples of generic command lines in SPI format.

Example #1: We have an executable (process1.exe) simply taking two command-line parameters. Supposing that these parameters are symbolized by the dataslots *in1* and *in2* of the process, the generic command line is written as follows:

```
process1.exe %in1 %in2
```

Example #2: The C compiler gcc accepts many command-line options (e.g. the output file specification) marked by special tokens (“-o”). A generic command line for this tool could look like this one:

```
gcc -o %output %source_file
```

Please note also that the **output** value of the process (the compiled object file) is here passed (as a file path) as an **input** parameter of the tool, in order for it to know where to store the produced file.

Example #3: This process accepts a variable number of input values, i.e. a list of files to join. This is symbolized at the process level by one dataslot having “multi” cardinality (*in*), which is referenced in the generic command line using the (%*in*) token, ensuring that the script will receive as many command-line parameters as there are elements in the list in:

```
joinX2.ksh (%in) > %out
```

As this process produces its output on the standard output channel, we need to redirect it to the file path associated to the process’ output dataslot (*out*) using the shell operator ‘>’.

## External process guidelines

All SAAM processes integrated into SIM using a *.spi* file are handled the same generic way by SIM. The same algorithms are used to analyze the process interface, check its input, generate its command line, launch it, and gather its output once it is completed. This is why these external processes must conform to the present guidelines.

### Calling syntax

The calling syntax of the process must be compatible with a representation in the *cmdLine* attribute of the SPI format.

### Output specification

The output file names of the process must be passed as a command-line parameter of the process.

*Rationale:* this allows the SIM engine to decide by itself the name and location of each output file of the process before executing it, and to know where to find these outputs when the process has exited.

### Return code

The process should return an exit code equal to 0 if it was completed successfully or a non-zero exit code if it has encountered an error.

*Rationale:* the exit code of the process is retrieved by SIM, which issues a warning if it is

different of 0. NB: an error at process level constitutes only a warning at diagram level, because even if a process has crashed, other parts of the diagram may be working correctly.

### Standard output

Data produced on the standard output channel by the process is usually considered as log information, but in some cases it can also be considered as an output of the process, so that it will be handled by SIM. To that end, the command-line construction string must include the redirection of the standard output to an output file value, e.g. "script.ksh %input1 %input2 > %output".

### Standard error

The process should produce data on the standard error channel only if it has encountered an error.

*Rationale:* The contents of the standard error channel of a process are stored in a "stderr.log" file for each process execution. A warning is issued in SIM if that file is not empty.

### Input file types

The process should not try to read the extensions of the input files.

*Rationale:* The file names used internally by SIM are generated without extension.

## Distribution

Each external process integrated in SIM is associated to its own SPI file. All these SPI files are intended to be installed in the process folder.

To install a new SPI process on SIM, one must:

- Copy the executable/script and its dependencies in the SAAM *bin* folder or in a folder which is in the PATH environment variable.
- Copy the associated *.spi* file in the SPI folder.

*Note:* the location of this folder can be view from SIM preference dialog

### 4.15.2.2 Plug-in processes

Plug-in SIM processes are implemented in C++ and, as opposed to built-in processes, are not implanted within the *SimCore* library, but in other separate dynamic libraries (DLLs).

Plug-in processes can also have dedicated dialog boxes for edition of the non-linkable parameters.

Integrating a new plug-in process in SIM is done by creating a new Qt Library project, then adding in it a class deriving from the *SimCore::PlugInProcess* class and implementing or reimplementing some key methods, which are detailed below.

A motivation of plug-in processes is to be able to use the same DLL module:

- In SIM, for use in diagrams.
- In a separate launcher, for use in SAAM.

## Project configuration

The new project must be linked to the *SimCore* library. To do that, the following project properties need to be modified:

- **Additional Include Directories:** The directory containing the SimCore header files (e.g. C:\SAAMlib\simcore1\_o\include) must be added to the list.
- **Additional Library Directories:** The directory containing the SimCore binaries (e.g. C:\SAAMlib\simcore1\_o) must be added to the list.
- **Additional Dependencies:** SimCore\_1\_o.lib must be added to the list.

## Cloning ability

The clone() method must be implemented for a plug-in process. The general principle of this method is to create a copy of the object on which the method is called, and return the address of the new instance, but as a pointer on a *GenericProcess*. This genericity of the returned value allows to copy any instance of *GenericProcess* or a derived class without knowing exactly which class the copied instance belongs to. This is mandatory for features such as the copy/paste function and the process toolbox.

The clone() method can be generally written the following way:

```
GenericProcess* T::clone() const
{
    return new T(*this);
}
```

Where “T” is to be replaced with the name of the derived class, as shown in the following example (taken from the class *ProcessAddition*):

```
GenericProcess* MyPluginProcess::clone() const
{
    return new MyPluginProcess(*this);
}
```

Please note that, although the return type of the method is *GenericProcess\**, the object instantiated is indeed a new instance of the class *ProcessAddition*, by calling the copy constructor (explicit or implicit) on the object *this*.

## Declaring the SIM process’ appearance and interface

A *initSimInterface()* virtual method is declared in the *PlugInProcess* class. Plug-in processes developers should put the appearance and process declaration code in the implementation of this method. That method will be called immediately after the plug-in has been loaded and constructed by SIM, ensuring that its interface is defined at any time during SIM execution.

The following parameters should be initialized:

- The ID of the plugin: The ID must be unique among every SIM process.
- The caption: What appears in the block in SIM
- The Author (optional): What appears in the "Author" field in SIM.
- The Description (optional): What appears in the "Description" field in SIM.
- The dataslots: The inputs/outputs of the process.

*Example:*

```

MyPluginProcess::initSimInterface()
{
    //ID of the plugin (must be unique)
    setId("SIM_PLUGIN_DEMO");

    //Caption: what appear in the block in SIM
    setCaption("SIM_PLUGIN_DEMO");
    setAuthor("Eurocontrol");
    setDescription("Demo plugin");

    // Dataslots
    addDataSlot("browser1", "Input File",      "FILE", "ASE", SINGLE, INPUTSLOT);
    addDataSlot("value",    "Unused INT",      "PRIMARY", "INT", SINGLE, INPUTSLOT)->set
    addDataSlot("browser2", "Output File",      "FILE", "SO6", SINGLE, OUTPUTSLOT);
}

```

The ID of the dataslot (the first parameter of the `addDataSlot` method) is very important. It will be used later to retrieve the value of a parameter from the SIM ParamStates. So you should choose carefully a short and understandable name.

## Process configuration

The `_hasConfigurationDialogBox` member variable of *PlugInProcess* can be set to true if the plug-in process class provides a dedicated dialog box for parameter edition.

The *PlugInProcess::configure()* method can then be reimplemented to handle the management of that dialog box, usually conforming to the following pattern:

1. Get current state (linkable or not) and values of the process' dataslots.
2. Instantiate the dialog box.
3. Fill the dialog box widgets with values from the dataslots.
4. Show dialog box (modal state); the user can then change the values.
5. If changes validated by the user, extract them from the dialog box
6. Reapply the values from the dialog box to the process.

Example taken from a “Compare Network” plug-in:

```

void MyPluginProcess::configure(QObject* parentWidget)
{
    QMap<QString, ParamState> paramStates = GenericProcess::getParamStateList();

    MyDialog* dialog = new MyDialog;
    dialog->loadParamStates(paramStates);

    if (dialog->exec() == QDialog::Accepted)
        GenericProcess::setParamStateList(dialog->getParamStates());

    delete dialog;
}

```

The *ParamState* class is designed to hold the characteristics of a dataslot that are useful for the configuration dialog box: the dataslot ID, its value (if any) and its linkable/non-linkable state.

The *getParamStateList()* and *setParamStateList()* methods of *GenericProcess* allow respectively to extract a set of *ParamState* from the dataslots of a process and to apply it back to the process (steps 1 and 6).

What must be developed for each plug-in is the mechanism to fill the dialog box with the set of *ParamState* before displaying it and extract them back after the user closed the box.

In the present example, the original the class `MyDialog` have to provide the methods *loadParamStates()* and *saveParamStates()* to handle that logic.

## Implementing the process' execution

Similarly to built-in processes, the *execute()* method needs to be reimplemented.

One strong constraint applies for this implementation: do not create *QWidgets* or one of its derived classes. The main reason standing behind that rule is because within SIM, the execution of a diagram takes place in a separate thread. *QWidgets* or derivatives can only be instantiated in the GUI thread (i.e. main thread).

*Example:*

Execution of a process which add two integer. For convenience, some explicitly named pointers to the four dataslots of the process are declared:

```
GenericProcess::ExitStatus ProcessAddition::execute()
{
    DataSlot* slotA = getSlot(0);
    DataSlot* slotB = getSlot(1);
    DataSlot* slotRes = getSlot(2);
    DataSlot* slotRestxt = getSlot(3);
```

Input values are extracted from input dataslots:

```
int a = slotA->getValue().toInt();
int b = slotB->getValue().toInt();
```

The computation of the result ( $a + b$ ) is simple enough to be directly integrated in the code storing the values in the output dataslots, shown below:

```
slotRes->setValue(QVariant(a+b));
slotRestxt->setValue(QVariant(QString("%1 + %2 = %3").arg(a).arg(b).arg(a+b)));
```

Everything went as expected; the normal exit code is returned:

```
return GenericProcess::NormalExit;
}
```

## Optional methods

Some additional data can be re-implemented to customize the plug-in:

- `isReadyToExecute()`
- `dataSlotModified()`
- ...

Read the SIM technical documentation to get more information.

## Qt plugin declaration

The declaration of a Qt plugin needs additional code lines.

In the cpp file export the plugin with the following Qt command:

```
Q_EXPORT_PLUGIN2(MyPlugin, MyPluginProcess)
```



*Note:* In this example MyPlugin is the name of the dll and MyPluginProcess in the name of the SIM process class which derive *SimCore::PlugInProcess*.

Finally, in the header file, call the following Qt macro inside the class definition:

```
Q_OBJECT  
Q_INTERFACES( SimCore::PlugInProcess )
```

## Distribution

Plug-in processes are compiled to DLL files. These DLLs should simply be stored to the Plug-ins folder designed for that purpose.

*Note:* The location of the this folder can be view from SIM preference dialog

### 4.15.2.3 How to choose the appropriate integration technique

As seen in the previous sections, several techniques are offered for a developer to add a new process block available for use in SIM. Choosing the appropriate technique among these three depends essentially on the “building bricks” already available when developing the new SIM process. The developer should evaluate the different alternatives in increasing difficulty order, which is:

1. SPI process
2. SIM diagram
3. Plug-in process

The first question that must be answered by the developer developing a new SIM process is:

**Q1:** *Is there an existing command-line executable or script that implements the needed feature? Does it comply with the SPI guidelines?*

If the answer is yes, then the developer will simply write and distribute a SPI file corresponding to that external tool. If the answer is no, the next question is:

**Q2:** *Is it possible to implement the needed feature in a SIM diagram?*

If the answer is yes, then the new SIM process will be implemented as a SIM diagram and added to the set of installed SAAM diagrams. If no, then the process will be implemented in C++ as a plug-in process.



# Chapter

---



5

## CHAPTER 5: Appendices

### 5.1 Command Line

SAAM can be launched from a shell or a DOS command prompt

SAAM command line options:

`[path_of_saam]saam [-h] [-az] [-fs] [-p[2|3]] [-2D] [[path]file.tdv]`

`/h` or `-h` for this help message.

`/az` or `-az` for autozoom: zoom, centre, tilt (top view) and rotate (north oriented) the map based on airspace objects present in the map.

`/fs` or `-fs` for full screen display.

`/p[2|3]` or `-p[2|3]` for printing in a file: the map is printed, full screen without splash screen in a png file located in the same directory as the TDV. Default print quality with `-p`, enhanced with `-p2`, and best with `-p3`. Once printed, SAAM quit.

`/2D` or `-2D` for 2D display: the map is forced to be displayed in 2D.

`/r` or `-r` to Run silently pre-defined SAAM User Task List, once TDV file is loaded. When executed SAAM quits. Might be incompatible with printing facility.

`[path]file.tdv`: SAAM load and display this TDV file, else the default one is used.

### 5.2 File Types

The following are a list of file types used by SAAM. All SAAM files are text files (ASCII) and can be edited with a standard text editor, if needed.

#### 5.2.1 aco - Airspace Colour file

Airspace COlour				
extension	aco			
origin	SAAM			
separator	blank			
sort	no sort			
comment	Describe several named display features and colour for a set of 3D airblocks			
#	Field	Type	Size	Comment
1	version	num	~	version of format (=1)
2	name of airblock	char	~	name can be: sector:airblock
3	airblock low level	num	3	expressed in Flight Level. CAUTION: padded with 0 !!!
4	airblock top level	num	3	expressed in Flight Level. CAUTION: padded with 0 !!!
5	number of colours	num	~	give the number of following different colours associated to the airblock
6	version of colour	num	~	version of colour (=1)
7	name of colour	num	~	identify the name of the colour (like "overload"). "?" is the default colour name
8	feature of colour	num	~	indicate the way to display the colour (transparent, top solid, ...), see below
9	colour flag	num	1	1 mean the colour is described here, 0 means no colour (comes from TDV)

#	Field	Type	Size	Comment
10	colour type coding	num	1	"C"
11	transparency of colour	num	~	min 0, max 255
12	red of colour	num	~	min 0, max 255
13	green of colour	num	~	min 0, max 255
14	blue of colour	num	~	min 0, max 255

### Field 8, feature of colour code:

indicates if the features for this airspace come from the TDV line (common to all airspace of the file) or from the airspace itself. Each feature has 3 values (except flight level):

0 means the feature comes from TDV line

1 means the feature comes from the airspace itself and is set to ON

2 means the feature comes from the airspace itself and is set to OFF

bit 0 and 1: top flag

bit 2 and 3: lighting flag

bit 4 and 5: transparent flag

bit 6: bottom level flag (0 means come from TDV if possible (!=999), 1 means come from the airspace header field 5)

bit 8: top level flag (0 means come from TDV if possible (!=999), 1 means come from the airspace header field 6)

bit 10: label flag (0 means come from TDV if possible (!=9), 1 means label is displayed, 2 label is not displayed (see also content of the label))

bit 12: soft lock flag (0 means come from TDV if possible (see lower/upper case of the airspace file name in TDV volume format), 1 mean soft lock, 2 means no soft lock (normal))

### Examples:

192: means only bottom and top level are set from the airspace itself, the rest comes from TDV line

208: means bottom and top level and transparent ON for the airspace itself, rest is from TDV

224: means bottom and top level and transparent OFF for the airspace itself, rest is from TDV

### Example:

```

1 LIPPMB1:A000165 000 085 1 1 ? 192 1 C 64 214 229 202
1 LIPPMB1:A000166 000 085 1 1 ? 192 1 C 64 214 229 202
1 LIPPMB1:A000167 000 085 1 1 ? 192 1 C 64 214 229 202
1 LIPPMB1:A000168 085 135 1 1 ? 192 1 C 64 214 229 202
1 LIPPMB1:A000169 085 135 1 1 ? 192 1 C 64 214 229 202
1 LIPPMB1:A000170 135 175 1 1 ? 192 1 C 64 214 229 202
1 LIPPMB1:A000171 135 175 1 1 ? 192 1 C 64 214 229 202

```

## 5.2.2 Aircraft Performance Profile Map

Aircraft Performance Profile Map				
extension	map.txt			
origin	SAAM performance map file			
separator	whitespace			
sort	no sort			
comment	For aircraft profile processing, used with "Aircraft Performance Data". Contains link between aircraft type and family			

#	Field	Type	Size	Comment
First line				
1	file format id			Version of the file format, formatted as n.n
2	generation date	date		Date at which the file was generated, formatted as dd/mm/yyyy.
3	comment	char	till eol	A comment field. Normally contains indications on the origins of the data.
Second line				
#	Field	Type	Size	Comment
1	map count	num		Number of mappings in the file.
Mapping Entry				
#	Field	Type	Size	Comment
1	aircraft id	char		An aircraft id code followed by SR (or LR if multiple profiles are given for a single aircraft type). SR means Short Range, LR Long Range
2	performance index	num		Base 1 index in the performance data file. Indicates which performance profile family is associated for this aircraft type.

### Example (1 flight 10 first lines of body):

```

3,3 20-08-2003 Based on BADA 3.4 + CFMU V31
2999
32SR 161
35SR 161
37SR 161
39SR 161
A109SR 162
A10ASR 163
A10LR 118
A10SR 117
A119SR 162

```

## 5.2.3 Aircraft Performance Profile Data

Aircraft Performance Profile Data	
extension	dat.txt
origin	SAAM performance data file
separator	whitespace
sort	
comment	For aircraft profile processing, used with "Aircraft Performance Map". Contains Family performance. Note: reference to performance is made via index number (first perf has index 1...) see Aircraft Performance Map

#	Field	Type	Size	Comment
<b>First line</b>				
1	file format id			Version of the file format, formatted as n.n
2	generation date	date		Date at which the file was generated, formatted as dd/mm/yyyy.
3	comment	char	till eol	A comment field. Normally contains indications on the origins of the data.
<b>Second line</b>				
1	profile count	num		Number of profiles in the file.
<b>HEADER</b>				
1	name	char	4	NOT an ID, facilitate searches (generally first aircraft name having this perf)
2	nb line	num		total number of following body lines
3	nb_climb_line	num		total number of following climbing lines
4	nb_line_descent	num		total number of following descend lines
5	optimum_FL	num		optimum' flight level (never higher than max flight level given by the table)
<b>BODY</b>				
1	length	num		length expressed in NM from departure or to arrival (goes from 0 to N, step 1)
2	level climb	num		climbing level at distance 'length' from departure
3	level descent	num		descent level at distance 'length' from arrival
4	time climb	float		time in minute when climbing to reach NM length from departure
5	time descent	float		time in minute when descending from top of descent to arrival
<b>Cruise speeds</b>				
1	size	num		Number of entries
2	level	num		FL
+2n				
3	cruise speed	num		speed in knots at 'level'
+2n				

**Example (1 flight 10 first lines of body):**

```

3,3 20-08-2003 Based on BADA 3.4 + CFMU V31
183
A306___.PTF 168 167 122 370
0 0 0 0 23.2341
1 8 4 0.3 22.9341
2 15 7 0.6 22.6341
3 23 11 0.9 22.3341
4 30 14 1.2 22.0341
5 38 18 1.5 21.7341
6 45 22 1.8 21.4341
7 53 25 2.1 21.1341
25 0 0 5 0 10 0 15 0 20 0 30 230 40 233 60 272 80 280 100 289 120 297 140 306 160 389 180 401 200
-> 413 220 425 240 438 260 452 280 466 290 468 310 464 330 459 350 455 370 453 390 453

```

## 5.2.4 are - Newmaxo ASCII Region file

Newmaxo ascii region file	
extension	are
origin	newmaxo, SAAM, CAPAN
separator	blank
sort	Body is sorted by point sequence (for SAAM it should be preferably clockwise to determine inside from outside)
comment	Describe a 2D/3D piece of airspace. Has a header and a body. For SAAM, normally linked to an "sls" file see Converting SAAM format to/from Gasel format

#	Field	Type	Size	Comment
1	nb_point	num	~	contains the number of lines (=vertices) of the following body
2	latitude	num	~	in minutes (decimal for SAAM), location of the label
3	longitude	num	~	in minutes (decimal for SAAM), location of the label
4	flights	num	~	value 1 (can be negative), can be 0
5	bottom_level	num	~	low level of the volume in flight level (FL) (see below), can be 0
6	top_level	num	~	high level of the volume in flight level (FL) (see below), can be 0
7	surface	num	~	value 2 (can be negative), can be 0
8	sector_num	num	~	value 3 (can be negative), can be 0
9	flight time	num	~	value 4 (multiplied by 100 to get 2 decimals, can be negative), can be 0
10	traffic density	num	~	value 5, can be 0
11	x mileage	num	~	Label content for SAAM (see below), can be 0
12	rte extens.	num	~	Feature code for SAAM (see below), can be 0
13	value 1	num	~	Color code for SAAM (see below), can be 0
14	value 2	num	~	value 6, can be 0
15	name	char	~	name or code of the volume (max 12 char for newmaxo, max 24 character for SAAM) (see below)
<b>Body, contains vertices coordinates (the polygon must be closed: first point = last point)</b>				
1	latitude	num	~	in minutes (in minutes decimal for SAAM)
2	longitude	num	~	in minutes (in minutes decimal for SAAM)

### Additional comments:

Field 5 & 6: low/high level: if low and high levels are swapped, the volume is marked negative (to be subtracted)	
Field 11, label content code	Indicates the content of the label:
	bit 0: sector name display (if only sector is display then only one airblock of the sector will be chosen for display)
	bit 1: airblock name display (all airblock have a label)
	bit 2: min/max FL display (of the airblock by default or of the sector if sector displayed)
Field 12, feature code	bit 3 to 7: value display (bit 3=value1, bit 4=value2, bit 5=value3, bit 6=value4, bit 7=value5)
	indicates if the features for this airspace come from the TDV line (common to all airspace of the file) or from the airspace itself. Each feature has 3 values (except flight level):
	0 means the feature comes from TDV line
	1 means the feature comes from the airspace itself and is set to ON
	2 means the feature comes from the airspace itself and is set to OFF
	bit 0 and 1: top flag
	bit 2 and 3: lighting flag
	bit 4 and 5: transparent flag



	<p>bit 6: bottom level flag (0 means come from TDV if possible (!=999), 1 means come from the airspace header field 5)</p> <p>bit 8: top level flag (0 means come from TDV if possible (!=999), 1 means come from the airspace header field 6)</p> <p>bit 10: label flag (0 means come from TDV if possible (!=9), 1 means label is displayed, 2 label is not displayed (see also content of the label)</p> <p>bit 12: soft lock flag (0 means come from TDV if possible (see lower/upper case of the airspace file name in TDV volume format), 1 mean soft lock, 2 means no soft lock (normal))</p> <p>Examples:</p> <p>192: means only bottom and top level are set from the airspace itself, the rest comes from TDV line</p> <p>208: means bottom and top level and transparent ON for the airspace itself, rest is from TDV</p> <p>224: means bottom and top level and transparent OFF for the airspace itself, rest is from TDV</p>
Field 13, color code	<p>If this field is set to 0 (ZERO) is means the color for this airspace is coming from the TDV line, else it represents the value of the color for the airspace itself.</p> <p>The coding is (for each component varying between 0 and 255):</p> $\text{red} * 16777216 + \text{green} * 65536 + \text{blue} * 256 + \text{transparent}$ <p>look at color transformation in this excel file.</p> <p>If level of transparency is 0 (whatever the values for R,G &amp; B are) then the level of transparency is taken from TDV file.</p> <p>The most black most transparent (=totally translucent) for an individual airspace is then: 16843009.</p> <p>The most black less transparent (=totally black) for an individual airspace is then: 16843263.</p> <p>Try the 2 "color transformation" tables on the left side.</p>
Field 15, name convention	<p>The name <u>might</u> have 2 parts, separated by a semi-colon.</p> <p>If the name have 2 parts separated by semi-colon (this happened when ".are" is alone, so no associated sls), in that case, the first part is the name of the sector (or the group), the second part is the name of the airblock (or piece of airspace which is the element that belongs to the group) described by the data itself.</p> <p>Example: sEBBRTMA:o36EB</p> <p>If the name has one part, (an associated ".sls" should exist) it might represent the name of the airblock which is used in the SLS file, in that case it <b>MUST</b> be the same name between ARE an SLS !!!</p> <p>Example: LFO34</p>

COLOR TRANSFORMATION	
color input:	4294901862
red result:	255
green result:	255
blue result:	0
trans. result:	102
red input:	255
green input:	255
blue input:	10
trans. Input:	102
color result:	4294904422
FEATURE CODE TRANSFORMATION	
feature code input:	
top flag result:	
lighting flag res.:	
trans. flag res.:	
bottom lev. flag res.:	

top lev. flag res.:	
label flag result:	
soft lock flag res.:	
top flag input:	
lighting flag inp.:	
trans. flag inp.:	
bottom lev. flag inp.:	
top lev. flag inp.:	

**Example:**

```

14 2799 925 0 0 660 0 0 0 0 0 0 0 0 0 LJ
2.799.925
2792 932
2797 943
2785.91 954.62
2780 942
2774 947
2781 957
2784 964
2784 975
2784 977
2790 974
2794 982
2790 992
2799 925

```

**5.2.5 arp - Airport**

Airport definition	
extension:	arp
origin:	manual input
separator:	blank
sort:	no
comment:	defines the geographical location of airports used in the assignment process.

NORMAL LINE				
#	Field	Type	Size	Comment
1	airport name	char	4	ICAO airport name
2	latitude	num	~	in minutes (in minutes decimal for SAAM)
3	longitude	num	~	in minutes (in minutes decimal for SAAM)
4	FIR name	char	5/7	FIR name associated to the airport, with "FIR" as a suffix. This field is optional. "NOFIR" is used when no FIR information is available.

**Example:**

```

EGSX 3103.267 9.350 EGTTFIR
EGSY 3203.650 -83.300 EGTTFIR
EGSZ 3620.000 -102.000 EGPXFIR
EGTA 3106.883 -56.550 EGTTFIR

```

```
EGTB 3096.700 -48.483 EGTTFIR
```

## 5.2.6 as4

AS4				
extension	as4			
origin	SAAM filter so6ToAs4			
separator	blank			
sort	yes: on "time begin segment" then on "time end segment"			
comment	ascii file used as an input in the SAAM/3Dviewer for traffic animation			

#	Field	Type	Size	Comment
1	fligth key	num		could be used to retrieve information on the flight (not used)
2	sequence	num		give the sequence of segment route for a given flight (not used)
3	segment name	char	max 11	pointName1_pointName2 (currently ignored)
4	lat begin segment	float		in minute decimal
5	lon begin segment	float		in minute decimal
6	FL begin segment	float		in flight level with decimal
7	time begin segment	float		in second decimal
8	lat end segment	float		in minute decimal
9	lon end segment	float		in minute decimal
10	FL end segment	float		in flight level with decimal
11	time end segment	float		in second decimal

### Example:

```
13972 15 VALPO_%LI7 2750.1 685.9 350 0 2754.5 678.35 350 1.12
16185 9 $aLYh_%ES21 3582.7 899.55 350 0 3585.2 849.8 307 1.13
11907 19 %LK6_VLM 2969.1 913.3 191.2 0 2982.7 904.6 142.6 2
4912 4 $adEt_REDFA 3123.4 129.2 290 0 3127.7 149.7 290 2
7754 3 $afKv_RASDA 1963.9 1857.8 350 0 1986.7 1856.6 350 3
7876 10 $afPA_DVR 3074.3 56.4 370 0 3070.5 82.7 333.2 3.2
13968 20 %ED88_%ED15 2979.7 506 350 0 3002 447.81 350 5
2441 19 PON_EVX 2945 122.1 330 0 2941 73.9 330 5
4888 7 EXMOR_BHD 3070 -201 370 0 3023 -209 370 6
3277 12 AGN_ANETO 2633.4 52 370 0 2562 33.8 370 9
```

## 5.2.7 ase - ASCII Segment file

Ascii SEgment				
extension	ase			
origin	so6ToNetwork SAAM module or SAAM Network editor			
separator	blank			
sort	No sort for SAAM, NEWMAXO needs to have the number of flights (first field) ordered by increasing values			
comment	Describe a route network, each line represent a route segment.			

#	Field	Type	Size	Comment
1	flight count	float		normally number of flights using this route segment, could be a load
2	segment parity	num		0=NO, 1=ODD, 2=EVEN, 3=ODD_LOW, 4=EVEN_LOW, 5=ODD_HIGH,

				6=EVEN_HIGH
3	segment type	num		0=NO, 1=NORMAL, 2=ARRIVAL, 3=DEPARTURE (permanent rte segment) 20=NO, 21=NORMAL, 22=ARRIVAL, 23=DEPARTURE (CDR Generic) 40=NO, 41=NORMAL, 42=ARRIVAL, 43=DEPARTURE (CDR 1) 60=NO, 61=NORMAL, 62=ARRIVAL, 63=DEPARTURE (CDR 2) 80=NO, 81=NORMAL, 82=ARRIVAL, 83=DEPARTURE (CDR 3) 100=NO, 101=NORMAL, 102=ARRIVAL, 103=DEPARTURE (CDR 1+2) 120=NO, 121=NORMAL, 122=ARRIVAL, 123=DEPARTURE (CDR 1+3)
4	lat begin segment	float		in minute decimal (only in minute for Newmaxo)
5	lon begin segment	float		in minute decimal (only in minute for Newmaxo)
6	lat end segment	float		in minute decimal (only in minute for Newmaxo)
7	lon end segment	float		in minute decimal (only in minute for Newmaxo)
8	segment name	char	max 11	routePointNameBegin_routePointNameEnd (separator can be "_" or "-", but the first one is preferred)

**Example:**

```

373 2 3 2941.1 145.9 2919.25 119.02 WDG_RBT
376 5 1 2825.3 466 2831 480 BERSU_RIVEL
384 2 1 3049 505 3014 556 %ED24_GELNI
387 5 41 2859 399 2877 400 ARPUS_MIRGU
445 2 1 3078 36 3070.85 82 DET_DVR

```

**Note:** segment ARPUS\_MIRGU is a CDR

## 5.2.8 asx - ASCII Segment Extended file

Ascii Segment eXtended	
extension	asx
origin	so6ToNetwork SAAM module or SAAM Network editor or CFMU data translator
separator	blank
sort	No sort
comment	Describe a route network (possibly loaded), each line represent a route segment or a single point or an airport

#	Field	Type	Size	Comment
1	flight count	float		normally number of flights using this route segment, could be a load
2	segment parity	num		0=NO, 1=ODD, 2=EVEN, 3=ODD_LOW, 4=EVEN_LOW, 5=ODD_HIGH, 6=EVEN_HIGH
3	item type	num		less than 900 means a SEGMENT being: ROUTE segments: 0=NO, 1=NORMAL, 2=ARRIVAL, 3=DEPARTURE (permanent ROUTE)

				20=NO, 21=NORMAL, 22=ARRIVAL, 23=DEPARTURE (CDR Generic ROUTE) 40=NO, 41=NORMAL, 42=ARRIVAL, 43=DEPARTURE (CDR 1 ROUTE) 60=NO, 61=NORMAL, 62=ARRIVAL, 63=DEPARTURE (CDR 2 ROUTE) 80=NO, 81=NORMAL, 82=ARRIVAL, 83=DEPARTURE (CDR 3 ROUTE) 100=NO, 101=NORMAL, 102=ARRIVAL, 103=DEPARTURE (CDR 1+2 ROUTE) 120=NO, 121=NORMAL, 122=ARRIVAL, 123=DEPARTURE (CDR 1+3 ROUTE) STAR segments: 200=NO, 201=NORMAL, 202=ARRIVAL (permanent STAR) 220=NO, 221=NORMAL, 222=ARRIVAL (CDR generic STAR) 240=NO, 241=NORMAL, 242=ARRIVAL (CDR 1 STAR) 260=NO, 261=NORMAL, 262=ARRIVAL (CDR 2 STAR) 280=NO, 281=NORMAL, 282=ARRIVAL (CDR 3 STAR) 200=NO, 201=NORMAL, 202=ARRIVAL (CDR 1+2 STAR) 220=NO, 221=NORMAL, 222=ARRIVAL (CDR 1+3 STAR) SID segments: 400=NO, 401=NORMAL, 403=DEPARTURE (permanent SID) 420=NO, 421=NORMAL, 423=DEPARTURE (CDR generic SID) 440=NO, 44
4	lat begin segment	float		in minute decimal
5	lon begin segment	float		in minute decimal
6	lat end segment	float		in minute decimal, equals to field 4 in case of single point or airport
7	lon end segment	float		in minute decimal, equals to field 5 in case of single point or airport
8	item name	char	max 11	routePointNameBegin_routePointNameEnd for segment or pointName or AirportName for single point. Letters not accepted are: ? / \ ' " ` ~ \$ # ^ & ( ) - + = { } [ ]   , < > space Special case of _ is used as a separator for segment
9	airway name	char		for instance UZ225, ? if no airway name
10	Lower Level Limit	num	3	3 digits expressed in FL
11	Upper Level limit	num	3	3 digits expressed in FL
14	airway seg. sequ.	num		only for airway: provide segments sequencing and possible break, starts at 1. Set to 0 for other items or other cases.
15	Layer Project ID	num		linked to Layer data file else 0

**Example (extract that do not correspond to any official data):**

```
373  2    3  2941.1  145.9  2919.25  119.02  BENSU_WOOD  UA47  295  999  1  0
```

## 5.2.9 cap - Capacity file

capacity	
extension	cap
origin	produced manually or by SAMAD loader or by the Airspace Config editor
separator	one of more spaces (ASCII character 32)
sort	no mandatory order (except for the default capacity - see Rule 8)
comment	each line contains the capacity for a elementary or collapse sector, possibly with an applicability time window

#	Field	Type	Size	Comment
1	sector name	char	26	the same one as the one found in field 1 of sls or found in the gsl or spc gasel files
2	sector capacity	int		0 till 999. 999 means infinite capacity
3	time window begin	char	5	expressed as HH:MM. 00:00 until 23:59
4	time window end	char	5	expressed as HH:MM. 00:01 until 24:00

### Rules

1. The capacity file does not contain applicability dates, only times. i.e. the capacities are valid for any date
2. fields 3 AND 4 together can be omitted, in such case the capacity represents the sector default capacity (e.g. the so-called declared capacity)
3. The default capacity of a sector can be overridden by a capacity applicable to a specific time window.
4. borders of a time window: "day time begin" is included in the time window, "day time end" excluded. In other words: "time window begin" >= time window < "time window end"
5. A day starts at 00:00 and ends at 24:00. (in accordance with note 3)
6. The "time window end" must be greater than the "time window begin"
7. Blank lines are allowed to better separate sectors from each others.
8. A sector cannot have several capacity rows that overlap in time. (Except with a default capacity). In particular, re-definition of a capacity (same sector & time) is forbidden. For example a sector default capacity cannot be redefined.
9. If a sector has several capacity figures (several rows), the rows do not need to be in chronological order, and do not need to be listed one after the other. Except for the default capacity (if any), that must be listed before any specific time window capacity.
10. If a sector does not have a default capacity row and its time window capacities do not cover the 24 hours, the capacity outside these time windows is unknown. Some SAAM modules might then use a general default capacity.
11. It is not required that the capacity file contains a capacity for all sectors (used in a SAAM study). Some SAAM modules might then use a general default capacity.
12. The capacity file, like any other SAAM data file, could contain data that do not comply to the above rules. (e.g. if a file is corrupted, if a user has modified the file with a text editor ...). For the capacity file, we could imagine that a time has no colon eg 2306 or is above 59 minutes eg 12:89 or that a sector has several time overlapping capacities. The module using the data should detect these errors, give a warning to the user or in a log, and, if possible and desirable, use a fall back position. For instance, a fallback position could be, in the case of a sector with several time overlapping capacity rows, 1) to keep only the first one, or 2) to change the time windows to remove the overlap. We let each module decide on the

Example 1:				Comment
LFEPPLOW	35	03:00	04:00	Capacity 35 ends just before 04:00
LFEPPLOW	42	04:00	06:00	Capacity 42 starts on 04:00 sharp!
				LFEPPLOW capacity outside 03:00 - 06:00 is unknown
EDDYLNO	35			35 is the default capacity applicable from 00:00 until 24:00
LIRRTUW	32	00:04	00:06	
LIRRTUW	36	07:00	08:00	
<b>Example 2: Forbidden syntax: time overlap</b>				
LFEPPLOW	35	03:00	04:00	
LFEPPLOW	38	03:30	04:30	forbidden: partial time overlap
EDDYLNO	35			
EDDYLNO	39			forbidden: 24 hours time overlap
<b>Example 3: default and specific capacity for the same sector</b>				
LFEPPLOW	35			Default capacity must always precede specific time window capacity
LFEPPLOW	39	03:30	04:30	
<b>Example 4: wrong ordering of rows (invalid)</b>				
LFEPPLOW	39	03:30	04:30	
LIRRTUW	36	07:00	08:00	
LFEPPLOW	35			Forbidden: default capacity after a specific one
<b>Example 5: a capacity for a 1 minute window</b>				
LIRRTUW	39	03:30	03:31	from 03:30:00 until 03:30:59.999999...

## 5.2.10 cdr - Conditional Routes file

CDR (conditional routes)				
extension	cor			
origin	SAAM network editor			
separator	semicolon ";"			
sort	no sort			
comment	<p>Each line defines a time and level window CLOSURE for a given CDR for a given scenario. All existing CDR of type 2 and 3 (by default closed) must be described in this file with at least one "default" scenario line.</p> <p>If "bottom level" equals "top level" equals 660, the CDR is OPEN for the given scenario.</p> <p>If a CDR exists and is not described in this file it is considered OPEN (case for CDR type 1 by default open).</p>			
#	Field	Type	Size	Comment
1	version	num		has value '1'
2	first point name	char	max 5	ICAO point name
3	second point name	char	max 5	ICAO point name

4	bottom level	num		division FL (values go from FL5 to FL5 with max 660, min 0)
5	top level	num		division FL (values go from FL5 to FL5 with max 660, min 0)
6	start time	char	5	format HH:MM padded with 0
7	end time	char	5	format HH:MM padded with 0
8	scenario name	char		specific "default" word is valid for any scenario not defined. Scenario name could also be a date with format "DD/MM/YYYY".

**Example:**

```
1;SUPAM;ARNEM;660;660;00:00;12:00;default
1;LOT;REMBA;0;660;00:00;24:00;default
1;LOT;REMBA;0;185;08:00;19:00;Any Friday
1;LOT;REMBA;185;285;10:00;12:30;Any Friday
1;SIGEN;TESGA;345;660;10:30;17:30;Any Friday
```

**Note:**

- SUPRA\_ARNEM is open by default,
- LOT\_REMBA is closed by default, it is partially open for "Any Friday" scenario
- SIGEN\_TESGA is not described in a "default" scenario so it is considered open outside "Any Friday" scenario

**5.2.11 cfg - Sector Configuration file**

sector CONFIGURATION	
version	1
extension	cfg
origin	gasel (=CFMU), SAAM
separator	semicolon ";"
sort	by family name then by config name
comment	

#	Field	Type	Size	Comment
1	family name	char		group within which all sectors of any config must be bounded, generally an ACC
2	config name	char		Generally a number or a string with a number indicating the number of sectors of the config
3	sector name	char		name of elementary sector or collapse sector can be used here

**Example with 2 centers (center EDBBACC has 2 configs; center EDFFEACC has 4 configs):**

```
EDBBACC;N2200;EDBBDBA
EDBBACC;N2200;EDBBFIT
EDBBACC;N2200;EDBBNORD
EDBBACC;N2200;EDBBSUEDOF
EDBBACC;NIGHT;EDBBDBA
EDBBACC;NIGHT;EDBBNORD
```



```

EDBBACC;NIGHT;EDBBSUED
EDFFEACC;CNFE1;EDFFOR
EDFFEACC;CNFE2;EDFFO45T
EDFFEACC;CNFE2;EDFFOR123
EDFFEACC;CNFE3A;EDFFOR12
EDFFEACC;CNFE3A;EDFFOR3T
EDFFEACC;CNFE3A;EDFFOR45
EDFFEACC;CNFE3B;EDFFO45T
EDFFEACC;CNFE3B;EDFFOR12
EDFFEACC;CNFE3B;EDFFOR3

```

### 5.2.12 cls - Sector file from CAPAN-like

Sector	
extension	.cls
origin	Capan-like
separator	blank
sort	no
comments	description of tasks and basic parameters for all sectors 4 types of record for each sector - the sector header - the basic parameters (6 lines) 2 type of records : those depending on conflict type and those which are general - the tasks description ( 1 task by line) only useful task are described (maximum 43) - the comments : each line starting with * is a comment

header of the sector				
#	Field	Type	Size	Comment
1	type name	char	5	SECTOR
2	sect num	int		num of current sector
3	sect code	char		sector code
4	low limit	int		low limit of the sector
5	up limit	int		upper limit of the sector
6	family name	char		family of the sector
7	available family	char		list of family of sector (yet is an option just for information)

#### Example

```

SECTOR    1    EDYCOHie          345    660          STANDARD    (STANDARD/SMALL/LARGE/TMA)

```

basic parameters depending on conflict type				
#	Field	Type	Size	Comment
1	type name	char		name of the basic parameter
2	value	int		value for conflict 1
3	value	int		value for conflict 2
4	value	int		value for conflict 3
5	value	int		value for conflict 4
6	value	int		value for conflict 5
8	value	int		value for conflict 6
7	value	int		value for conflict 7
8	value	int		value for conflict 8
9	value	int		value for conflict 9

**Example**

MON	1600	1600	1600	2000	2000	2000	2000	2000	2000
INT	800	800	800	1000	1000	1000	1000	1000	1000

**basic general parameters (not depending on conflict type)**

#	Field	Type	Size	Comment
1	type name	char		name of the basic parameter
3	value	int		

**Example**

MAXINT	3	MAXIMUM NUMBER OF RADAR INTERVENTIONS ACCEPTED FOR ONE FLIGHT IN ONE SECTOR
MAXSUP	6	MAXIMUM NUMBER OF RADAR SUPERVISIONS ACCEPTED FOR ONE FLIGHT IN ONE SECTOR
TIM	900	LONG FLIGHT TIME PARAMETER (SECONDS) 15 Minutes Default
SKP	-1	SKIP FLIGHT TIME (SECONDS)

**the tasks description ( 1 task by line)**

#	Field	Type	Size	Comment
2	type name	char	1	T for Task
3	No. Task	int		Num. of task
4	time EC	int		value in second for the executive controller
5	time PC	int		value in second for the planning controller

**Example**

T	1	0	1	Receipt of flight information
T	5	0	3	Receipt of a time and level estimate from a neighbouring ATC unit
T	15	0	3	Transmission of a time and level estimate to the next ACC / Verification
T	25	0	3	Receipt of a time and level estimate from the previous sector of the same

**5.2.13 clt - Task file from CAPAN-like**

Task file from CAPAN-like	
extension	.clt
origin	Capan-like
separator	blank
sort	no
comments	description of tasks and basic parameters for all family of sectors 3 types of record for each family <ul style="list-style-type: none"> <li>the basic parameters (6 lines) 2 type of records : those depending on conflict type and those which are general</li> <li>the tasks description ( 43 tasks , 1 by line)</li> <li>the comments : each line starting with * is a comment</li> </ul>

**Basic parameters depending on conflict type**

#	Field	Type	Size	Comment
1	family name	char		name of the family of sector
2	type name	char	3	name of parameter INT (Intervention) or MON (Monitoring)
3	value	int		value for conflict 1

Basic parameters depending on conflict type				
4	value	int		value for conflict 2
5	value	int		value for conflict 3
6	value	int		value for conflict 4
7	value	int		value for conflict 5
8	value	int		value for conflict 6
9	value	int		value for conflict 7
10	value	int		value for conflict 8
11	value	int		value for conflict 9

### Example

```
*****
*          BASIC PARAMETERS FOR STANDARD SECTORS
*****
STANDARD  MON   1600   1600   1600   2000   2000   2000   2000   2000   2000
STANDARD  INT    800    800    800    1000   1000   1000   1000   1000   1000
```

Basic general parameters (not depending on conflict type)				
#	Field	Type	Size	Comment
1	family name	char		name of the family of sector
2	type name	char	til 6	name of parameter
3	value	int		

#	Field	Type	Size	Comment
1	family name	char		name of the family of sector
2	type name	char	til 6	name of parameter
3	value	int		

### Example

```
STANDARD MAXINT 3 MAXIMUM NUMBER OF RADAR INTERVENTIONS ACCEPTED FOR ONE FLIGHT IN ONE S
STANDARD MAXSUP 6 MAXIMUM NUMBER OF RADAR SUPERVISIONS ACCEPTED FOR ONE FLIGHT IN ONE SE
STANDARD TIM 900 LONG FLIGHT TIME PARAMETER (SECONDS) 15 Minutes Default
STANDARD SKP -1 SKIP FLIGHT TIME (SECONDS)
```

The tasks description ( 1 task by line)				
#	Field	Type	Size	Comment
1	family name	char		name of the family of sector
2	type name	char	1	always = T for Task (T means : this line describes a task and not a basic parameter)
3	No Task	int		Num. of task
4	time EC	int		value in second for the executive controller
5	time PC	int		value in second for the planning controller

### Example

```
STANDARD T 1 0 1 Receipt of flight information
STANDARD T 5 0 3 Receipt of a time and level estimate from a neighbouring ATC unit
STANDARD T 15 0 3 Transmission of a time and level estimate to the next ACC / Verification of ACC
STANDARD T 25 0 3 Receipt of a time and level estimate from the previous sector of the same ACC
```

## 5.2.14 conf - Conflict file

Conflict	
extension	conf
origin	produced by snapshot.exe with So6 file
separator	blank
sort	no
comment	each line contains 2 conflicting flights. This file is an input for conflict analysis: clh, conf2tdp, densities ...

#	Field	Type	Size	Comment
1	key of flight 1	num		field 17 from so6
2	key of flight 2	num		field 17 from so6
3	time first contact	num		in second, within a day must process modulo 86400 of this value
4	conflict duration	num		in second
5	FL of flight 1 first contact	float		when first contact
6	FL of flight 2 first contact	float		when first contact
7	lat 1	float		location of flight 1 when first contact, in minute decimal
8	lon 1	float		location of flight 1 when first contact, in minute decimal
9	lat 2	float		location of flight 2 when first contact, in minute decimal
10	lon 2	float		location of flight 2 when first contact, in minute decimal
11	distance first contact	float		square nm, decimal
12	distance minimum	float		square nm, decimal
13	distance last contact	float		square nm, decimal
14	status 1 fist contact	num		same as so6 file: 0=climb, 1=descent, 2=cruise, used for type of conflict
15	status 2 fist contact	num		same as so6 file: 0=climb, 1=descent, 2=cruise, used for type of conflict
16	point 1 of seg 1	char		name of point, route segment where flight 1 is located
17	point 2 of seg 1	char		name of point, route segment where flight 1 is located
18	point 1 of seg 2	char		name of point, route segment where flight 2 is located
19	point 2 of seg 2	char		name of point, route segment where flight 2 is located
20	azimuth seg 1	float		degrees decimal, used to determine type of conflict: parallel, crossing or opposite
21	azimuth seg 2	float		degrees decimal, used to determine type of conflict: parallel, crossing or opposite
22	Time end of contact ?	float		in second
23	FL of flight 1 end contact	float		FL
24	FL of flight 2 end contact	float		FL
25	lat 1	float		location of flight 1 when end contact, in minute decimal
26	lon 1	float		location of flight 1 when end contact, in minute decimal
27	lat 2	float		location of flight 2 when end contact, in minute decimal
28	lon 2	float		location of flight 2 when end contact, in minute decimal
29	time of minimum distance	num		in second, within a day must process modulo 86400 of this value
30	FL of flight 1 min distance	float		Flight Level of Flight 1 when minimum separation distance is reached
31	status 1 min distance	num		same as so6 file: 0=climb, 1=descent, 2=cruise, used for type of conflict

32	azimuth seg 1 min distance	float		degrees decimal, used to determine type of conflict: parallel, crossing or opposite
33	FL of flight 2 min distance	float		Flight Level of Flight 2 when minimum separation distance is reached
34	status 2 min distance	num		same as so6 file: 0=climb, 1=descent, 2=cruise, used for type of conflict
35	azimuth seg 2 min distance	float		degrees decimal, used to determine type of conflict: parallel, crossing or opposite

**Example (1 line):**

```
105973632 105973659 1183077320 23 309.312500 317.047607 3016.445745 344.487062 3041.733769 348.262
... 645.253413 509.266157 750.890224 0 0 BULUX SOPOK PINUS LNO 135.40 208.62 1183077540 332.439026
... 3011.124959 388.244208 3016.377921 346.203506 1183077430 322.600006 0 104.68 325.545441 0 173.
```

**5.2.15 cos - Configuration Opening Scheme file**

configuration OPENING SCHEME	
version	1
extension	cos
origin	gasel (=CFMU), SAAM
separator	semicolon ";"
sort	by period name then by family name then by starting time
comment	

#	Field	Type	Size	Comment
1	period name	char		CFMU uses a date "DD/MM/YYYY". It could be a period name, like "BUSY".
2	family name	char		the same as the one found in "config" file
3	starting time	char		"HH:MM", config activation time beginning
4	ending time	char		"HH:MM", config activation time ending
5	config name	char		the same as the one found in "config" file
6	code	char		Today "E".

**Example:**

```
15/05/2004;LEBLAPP;00:00;05:29;CONF1;E
15/05/2004;LEBLAPP;05:30;19:29;CONF4W;E
15/05/2004;LEBLAPP;19:30;23:59;CONF1;E
15/05/2004;LECBACC;00:00;05:29;CNF2;E
15/05/2004;LECBACC;05:30;12:29;CNF10H;E
15/05/2004;LECBACC;12:30;19:29;CNF9D;E
15/05/2004;LECBACC;19:30;20:29;CNF8J;E
15/05/2004;LECBACC;20:30;21:59;CNF5;E
15/05/2004;LECBACC;22:00;23:59;CNF2;E
15/05/2004;LECLAPP;00:00;04:59;CONF1;E
15/05/2004;LECLAPP;05:00;07:29;CONF2A;E
15/05/2004;LECLAPP;07:30;19:29;CONF3A;E
15/05/2004;LECLAPP;19:30;21:59;CONF2A;E
15/05/2004;LECLAPP;22:00;23:59;CONF1;E
```

### 5.2.16 cost - Route Charge Cost file

Route Charge Cost	
version	1
extension	cost
origin	SAAM from Route Charge Module
separator	space
sort	by flight ID then by Country code
comment	Provide Route Charge Cost per flight per crossed country in EURO

#	Field	Type	Size	Comment
1	flight ID	num		Unique flight ID coming from input so6/t5 file
2	country code	char	2	ICAO code
3	cost	num		Route Charge Cost in Euro with at least 2 decimals (6 significant digits)

#### Example:

```

139729486 ED 356.159
139729486 LB 230.25
139729486 LC 86.7351
139729486 LH 148.107
139729486 LO 234.767
139729486 LT 365.319
139729486 LY 190.189
139729486 X2 19.3752
139729487 ED 48.1162
139729487 LH 181.893
139729487 LO 242.215
139729487 LR 385.217
139729487 UG 204.097
139729487 UK 262.747

```

### 5.2.17 crco - CRCO Intersection file

CRCO Intersection	
version	3
extension	crco
origin	SAAM from Route Charge Module
separator	blank
sort	flight id and chronological in flight
comment	Intersection summary with distance according to CRCO rules. Built on so6 and t5.

#	Field	Type	Size	Comment
1	country name	char		name is coming from T5 file sector name
2	callsign	char		
3	ICAO aircraft type	char		ICAO code
4	flight id	char		identifies flight in so6
5	entry lat	float		in minutes with 4 decimals
6	entry lon	float		in minutes with 4 decimals
7	exit lat	float		in minutes with 4 decimals

8	exit lon	float		in minutes with 4 decimals
9	distance	float		in kilometres with 8 decimals, 20km correction may have been applied
10	entry time	hhmmss		time of entry HHMMSS
11	entry date	yymmdd		date of entry YYMMDD
12	entry level	int		level of entry in FL with 3 decimals
13	exit time	hhmmss		time of exit HHMMSS
14	exit date	yymmdd		date of exit YYMMDD
15	exit level	int		level of exit in FL in FL with 3 decimals

**Example :**

```
ED N182QS B737 1523 3217.8167 599.3000 3018.8009 726.39836 396.03131430 090500 020322 0.000 095918 0203
LK N182QS B737 1523 3018.8009 726.3983 3019.0000 726.9864 0.78728520 095918 020322 270.000 095922 0203
LK N182QS B737 1523 3018.8009 726.3983 3019.0000 726.9864 0.78728520 111918 020322 270.000 111922 0203
ED N182QS B737 1523 3019.0000 726.9864 3217.8167 599.3000 395.93214016 111922 020322 270.000 121700 0203
```

**5.2.18 eep - Entry Exit Point**

EEP (Entry Exit Point)				
extension	eep			
origin	Manual user input to specify entry /exit points (gates) in a Free-Route project area			
separator	space			
sort	no sort			
comment	Each line defines the entry or exit points for a flow or a set of flows overpassing specific points. Blank lines are possible. It helps users to group flows into Free-routes "gates".			

#	Field	Type	Size	Comment
1	entry/exit tag	char		"I" for Input, "O" for output (this code is obsolete). Note: // is used for comment
2	flow origin	char		ICAO Airport name, can be "*" for any airport, a "?" replace any letter. Airport group can be created like: EDDF EDDL
3	flow destination	char		ICAO Airport name, can be "*" for any airport, a "?" replace any letter. Airport group can be created like: EDDF EDDL
4	entry/exit point name	char		entry/exit in free route area will be done via this point (it must exist)
5	route point names	char		list of point names, space separated, which individually gives a route condition to the flow to be routed to the entry/exit point

**Example:**

```
//DEP EDDF added by TCH 02/03/2005 (this is a comment)
I EDDF * RUDOT BITBU DIK MAKIK
I EDDF * DITAM ABAXA MASIR
O EDDF * KOLAG MIMVA RAVLO
```

**Note, how to interpret this example:**

```
Departures from EDDF must:
"- enter the area via RUDOT if they cross BITBU or DIK or MAKIK"
"- enter the area via DITAM if they cross ABAXA or MASIR"
"- exit the area via KOLAG if they cross MIMVA or RAVLO"
```

## 5.2.19 equi

equi	
extension	equi
origin	temporary files used during assignment (created in fap_2d_ZOD3.ksh)
separator	blank
sort	on flight ID numeric
comment	stored in "Cache" directory, contains link between flow ID and flight ID + expand info

#	Field	Type	Size	Comment
1	flow ID	num		initialised with first flight ID (smallest) of the flow
2	flight ID	num		as found in expand file
3	time departure	num	4	HHMM padded with 0
4	date departure	num	6	YYMMDD padded with 0
5	callsign	char		
6	origin	char	4	ICAO code
7	destin	char	4	ICAO code
8	aircraft type	char		
9	RFL	num		

### Example (10 flights / 7 flows):

```

41 41 1335 050630 PLK225 ULLI EKCH T154 340
51 51 1501 050630 RGL231 LPPT GMMN B190 210
61 61 1944 050630 RAM533 GMMN DIAP B738 350
101 101 0638 050630 LBC220 LATI LIRF B461 280
121 121 0638 050630 REA03CK EIDW EICK AT43 180
1061 1061 2224 050630 ADR650 LJJL LTBA A320 370
121 151 1043 050630 REA15CK EIDW EICK AT72 180
41 221 1334 050630 PLK225 ULLI EKCH T154 340
51 231 1454 050630 RGL231 LPPT GMMN B190 210
241 241 0217 050701 EXS6498 EDDK LIME A30B 290

```

## 5.2.20 exp2

Exp2 is a combination of Expand file (exp) and Flight Info file (flf)

Expand	
extension	exp
origin	traffic demand data
separator	semicolon ";"
sort	no
comment	input for SAAM modeling (assignment ...)

#	Field	Type	Size	Comment
1	origin	char		ICAO code (airport)
2	destination	char	4	ICAO code (airport)
3	not used	num	2	"00"
4	aircraft type	char	4	
5	RFL	num	3	Generally extracted from filled flight plan (not calculated)
6	zone origin	char	4	ICAO code (could be the same as airport)



7	zone destin	char	4	ICAO code (could be the same as airport)
8	flight ID	num		SAMAD ID or SAAM ID or ... (must be uniq)
9	date departure	num	6	YYMMDD
10	time departure	num	4	HHMM
11	time arrival	num	4	NOT USED = "!!!!"
12	callsign	char	till 7	
13	company	char	3	generally the 3 first letters of the callsign

**Example (10 flights):**

```

BGSF EKCH 00 B767 290 BGSF EKCH 367836 980626 1445 !!!! SAS292 SAS
BIKF EDDF 00 B757 370 BIKF EDDF 328474 980626 0745 !!!! ICE520 ICE
BIKF EDDK 00 CL60 290 BIKF EDDK 371896 980626 1516 !!!! GAF208 GAF
BIKF EDDK 00 B73S 270 BIKF EDDK 337460 980626 1740 !!!! ICE774 ICE
BIKF EDDL 00 B757 370 BIKF EDDL 346578 980626 0105 !!!! LTU1929 LTU
BIKF EDDM 00 B757 350 BIKF EDDM 351193 980626 0145 !!!! CMM759 CMM
BIKF EFHK 00 B73S 330 BIKF EFHK 328169 980626 0730 !!!! ICE342 ICE
BIKF EGLL 00 B757 290 BIKF EGLL 364422 980626 0750 !!!! ICE450 ICE
BIKF EGPF 00 B73S 290 BIKF EGPF 364457 980626 1200 !!!! ICE430 ICE
BIKF EGUN 00 C130 210 BIKF EGSS 366363 980626 0800 !!!! BLIGH49 ZZZ

```

Flight inFo	
extension	flf
origin	cfmu
separator	blank
sort	no
comment	input for SAAM traffic filtering and trajectories matching

#	Field	Type	Size	Comment
1	Flight SAAM ID	num		uniq for the day
2	Flight SAMAD ID	num		uniq for ever
3	TACT ID	num		uniq for ever, CFMU ID
4	SSR_CODE	char		
5	REGISTRATION	char		
6	Planned Date departure	num	6	YYMMDD padded with o's
7	Planned Time departure	num	4	HHMM padded with o's
8	ATFM DELAY	num		
9	REROUTING STATE	char		
10	most pen reg	char		if no regulation 'X' is present
11	type	char		letters indicating type of flight (military ?)
12	equipment	char		letters indicating equipment of flight
13	ICAO equipment	char		
14	COM equipment	char		
15	NAV equipment	char		
16	SSR equipment	char		
17	SURVIVAL equip.	char		
18	PERSONS ON BOARD	num		o means no information
19	top FL	num		
20	max RFL	num		

21	FLT PLN SOURCE	char		
----	----------------	------	--	--

**Example:**

```

1 118814379 0086774620080901 BB8093375720080901 @ RA86926 080901 0040 0 @ @ N SRWY @ @ S 340 340 FPL
2 118814380 0086257920080901 AA6817941420080901 @ OJAM 080901 0040 0 @ @ S SFHRWY @ @ S 370 370 FPL
3 118814381 0087092420080901 AA6818588420080901 @ N795AN 080901 0040 0 @ @ S SHIRWXYZ @ RNV1E2A1 SD 410 410 MFS
4 118814382 0086115020080901 BB8092860920080901 @ GFCLG 080901 0040 0 @ @ N SDHIRWXYZ @ RNP10 RNP5 S 380 380 FPL
5 118814383 0087102220080901 BB8093626320080901 @ SUGBE 080901 0040 0 @ @ S SDEPRWY @ @ S 370 370 FPL
6 118814384 0083939520080901 BB8092043720080901 @ DAXLF 080901 0045 0 @ @ N SDHIRWXYZ TCAS @ S 380 380 FPL
7 118814385 0086830220080901 AA6818393420080901 @ GBNLN 080901 0045 0 @ @ S SIPRWXYZ @ RNV1E2A1 RNP10 S 370 370 MFS
8 118814386 0087086220080901 BB8093614520080901 @ GBNLJ 080901 0045 0 @ @ S SIPRWXYZ @ RNV1E2A1 RNP10 S 370 370 MFS
9 118814387 0085611020080901 BB8092448320080901 @ TCOGJ 080901 0045 0 @ @ S SDEHILRWY @ @ S 340 340 FPL
10 118814388 0086824920080901 AA6818390020080901 @ FGLZI 080901 0045 0 @ @ S SJPRWXYZ @ RNV1E2A1 SD 370 370 FNM
11 118814389 0086215520080901 BB8092950520080901 @ GBYAO 080901 0045 0 @ @ S SEHIRWXYZ @ RNP10 S 360 380 FPL
12 118814390 0086258720080901 BB8092987920080901 @ ECKHU 080901 0045 0 @ @ S SJRWXYZ @ RNP10 C 380 380 FPL
13 118814391 0086298020080901 AA6817971720080901 @ PHSHL 080901 0045 0 @ @ N SGZ @ TCAS S 30 30 FPL
14 118814392 0086840920080901 BB8093430420080901 @ ECJVO 080901 0045 0 @ @ N SDGHRWY @ @ S 280 280 FPL
15 118814393 0087017720080901 BB8093575320080901 @ JYAIC 080901 0045 0 @ @ S SHIRWXY @ @ C 380 380 FPL
16 118814394 0086867220080901 BB8093449320080901 @ CSTGU 080901 0045 0 @ @ N SDHIRWXY @ @ C 370 370 FNM
17 118814395 0086749620080901 AA6818328220080901 @ PHBEF 080901 0045 0 @ @ S SDGHIJPRWXYZ @ RNAV1 RNAV5 RNP10 R S 350 350 MFS
18 118814396 0086825720080901 BB8093419220080901 @ TCJHA 080901 0045 0 @ @ S SDHIJPRWXY @ @ S 360 360 FPL
19 118814397 0086885320080901 BB8093464120080901 @ ZKNEV 080901 0045 0 @ @ S SGHIJPRWXY @ RNP4 RNP10 RNAV1 RNA SD 381 381 FPL
20 118814398 0086901520080901 AA6818445920080901 @ GCIVZ 080901 0045 0 @ @ S SIPRWXYZ @ RNV1E2A1 RNP10 S 350 370 MFS
21 118814399 0086690120080901 AA6818274320080901 @ DATUF 080901 0045 0 @ @ S SFGIRWXY @ @ S 370 370 FPL

```

**EXP2:**

	#E XP2	#Lo c	Field	Type	Comment
EXP	1	1	origin (ADEP)	char[4]	ICAO code (airport) International Civil Aviation Organisation
	2	2	destination (ADES)	char[4]	ICAO code (airport)
	3	3	not used	Numeric [2]	"oo"
	4	4	aircraft type	char[4]	
	5	5	RFL	Numeric [3]	Requested Flight level
	6	6	zone origin	char[4]	ICAO code (could be the same as airport)
	7	7	zone destin	char[4]	ICAO code (could be the same as airport)
	8	8	flight ID	Numeric	SAMAD ID or SAAM ID or ... (must be uniq)
	9	9	date departure	Numeric [6]	YYMMDD
	10	10	time departure	Numeric [4]	HHMM
	11	11	time arrival	Numeric [4]	NOT USED = "!!!!"
	12	12	callsign	char[<=7]	
	13	13	company	char[3]	generally the 3 first letters of the callsign
SEPARATOR	14	NA	Separator	Numeric [1]	value is o
UUID	15	NA	Universal Unique ID	char [ TODO ]	CALLSIGN-ADEP-ADES-EOBD-EOBT
	16	NA	Fips cloned	Alphabetic [1]	Y, N
SEPARATOR	17	NA	Separator	Numeric [1]	value is o
FLF	18	1	Flight SAAM ID	Numeric	uniq for the day
	19	2	Flight SAMAD ID	Numeric	uniq for ever
	20	3	TACT ID	Numeric	CFMU ID : Check with TACTID on ALL-FT

	21	4	SSR_CODE	char	plan identifiant (12bits mode A), Secondary Surveillance Radar
	22	5	REGISTRATION	char	
	23	6	Planned Date departure	Numeric [6]	YYMMDD padded with o's
	24	7	Planned Time departure	Numeric [4]	HHMM padded with o's
	25	8	ATFM DELAY	Numeric	Air Traffic Flow Management Delay : Difference between calculated by CASA take off time (CTOT) and estimated take off time (ETOT).;
	26	9	REROUTING STATE	char	
	27	10	most pen reg	char	if no regulation 'X' is present : Check with mostPenalizingRegulationId
	28	11	type	char	letters indicating type of flight (military ?)
	29	12	equipment	char	letters indicating equipment of flight
	30	13	ICAO equipment	char	International
	31	14	COM equipment	char	
	32	15	NAV equipment	char	
	33	16	SSR equipment	char	Secondary Surveillance Radar equipement
	34	17	SURVIVAL equip.	char	
	35	18	PERSONS ON BOARD	Numeric	o means no information
	36	19	top FL	Numeric	
	37	20	max RFL	Numeric	Requested Flight Level
	38	21	FLT PLN SOURCE	char	
SEPARATOR	39	NA	Separator	Numeric [1]	value is o
ALLFT	40	6	aobt	Numeric [14] YYYYMMDDHHMMSS	Actual Of Block in Time
	41	7	ifpsId	Alphabetic [2] Digit [8] XX12345678	Prisme (Integrated Initial Flight Plan Processing System)
	42	8	iobt	Numeric [14] YYYYMMDDHHMMSS	Initial Of Block in Time   FLF (planned time departure)
	43	9	originalFlightDataQuality	Alphabetic [3]	NON, PFD, RPL, FPL
	44	10	flightDataQuality	Alphabetic [3]	FLF (flt pln source) , NON, PFD, RPL, FPL
	45	11	source	Alphabetic [3]	UNK, FPL, RPL, AFI, MFS, FNM, AFP, DIV
	46	13	exemptionReasonType	Alphanumeric [4]	NEXE, EMER, SERE, HEAD, AEAP
	47	14	exemptionReasonDistance	Alphanumeric [4]	NEXE, LONG
	48	15	lateFiler	Alphabetic [1]	Y, N
	49	16	lateUpdater	Alphabetic	Y, N

			c[1]	
50	17	northAtlanticFlight	Alphabetic[1]	Y, N
51	18	cobt	Numeric[14]	Computed Of Block in Time YYYYMMDDHHMMSS
52	19	eobt	Numeric[14]	Estimated Of Block in Time YYYYMMDDHHMMSS
53	20	flightState	Alphanumeric[2]	NE, PL, PS, PR, SR, FI, FS, SI, TA, AA, CA, TE
54	21	previousToActivationFlightState	Alphanumeric[2]	NE, PL, PS, PR, SR, FI, FS, SI, TA, AA, CA, TE
55	22	suspensionStatus	Alphanumeric[2]	NS, ST, SM, RC, TV, NR, RV
56	23	tactId	Numeric[6]	FLF (tact id) (is it the same as tactid on FLF (3))
57	24	samCtot	Numeric[14]	YYYYMMDDHHMMSS
58	25	samSent	Alphabetic[1]	Y, N
59	26	sipCtot	Numeric[14]	YYYYMMDDHHMMSS
60	27	sipSent	Alphabetic[1]	Y, N
61	28	slotForced	Alphabetic[1]	Y, N
62	29	mostPenalizingRegulationId	Alphanumeric[8]	Is it the same as most pen reg on FLF (10)
63	30	regulationsAffectedByNrOfInstances	Numeric[4]	
64	31	excludedFromNrOfInstances	Numeric[4]	
65	32	lastReceivedAtfmMessageTitle	Alphabetic[3]	DES, ERR, FCM, FUM, FLS, REA, RFI, RJT, RRN, RRP, SAM, SIP, SLC, SMM, SPA, SRJ, SRM, SWM, UNK
66	33	lastReceivedMessageTitle	Alphabetic[3]	ABI, ACH, ACT, APL, ARR, CAN, CHG, CNL, DEP, DLA, ERR, EST, FPL, FSA, MFS, PAC, PFD, RPL, UNK
67	34	lastSentAtfmMessageTitle	Alphabetic[3]	DES, ERR, FCM, FUM, FLS, REA, RFI, RJT, RRN, RRP, SAM, SIP, SLC, SMM, SPA, SRJ, SRM, SWM, UNK
68	41	manualExemptionReason	Alphanumeric[1]	N, S, R
69	42	sensitiveFlight	Alphabetic[1]	Y, N
70	43	readyForImprovement	Alphabetic[1]	Y, N
71	44	readyToDepart	Alphabetic[1]	Y, N
72	45	revisedTaxiTime	Numeric[6]	000000..999999
73	46	tis	Numeric[6]	Time to Insert the Sequence , 000000..999999
74	47	trs	Numeric[6]	Time to Remove from Sequence , 000000..999999
75	48	toBeSentSlotMessageTitle	Alphabetic[3]	Related to flight progress , ABI, ACH, ACT, APL, ARR, CAN, CHG, CNL, DEP, DLA, ERR,

					EST, FPL, FSA, MFS, PAC, PFD, RPL, UNK
	76	49	toBeSentProposalMessageTitle	Alphabetic[3]	Related to flight progress , ABI, ACH, ACT, APL, ARR, CAN, CHG, CNL, DEP, DLA, ERR, EST, FPL, FSA, MFS, PAC, PFD, RPL, UNK
	77	50	lastSentSlotMessageTitle	Alphabetic[3]	Related to flight progress , ABI, ACH, ACT, APL, ARR, CAN, CHG, CNL, DEP, DLA, ERR, EST, FPL, FSA, MFS, PAC, PFD, RPL, UNK
	78	51	lastSentProposalMessageTitle	Alphabetic[3]	Related to flight progress , ABI, ACH, ACT, APL, ARR, CAN, CHG, CNL, DEP, DLA, ERR, EST, FPL, FSA, MFS, PAC, PFD, RPL, UNK
	79	52	lastSentSlotMessage	Numeric[14]	Related to flight progress , YYYYMMDDHHMMSS
	80	53	lastSentProposalMessage	Numeric[14]	Related to flight progress , YYYYMMDDHHMMSS
	81	54	flightCountOption	Alphabetic[1]	Indicates which flight plan should be / has been used when doing flight/count related operations. Used in TACT queries and replies , P, N
	82	55	normalFlightTactId	Numeric[6]	
	83	56	proposalFlightTactId	Numeric[6]	
	84	57	operatingAircraftOperatorIcaoId	Alphabetic[3]	
	85	58	reroutingWhy	Alphabetic[1]	N, M, C, A, O
	86	59	reroutedFlightState	Alphabetic[1]	P, E, T, R, V, N
	87	60	runwayVisualRange	Numeric[3]	
	88	70	ftfmAiracCycleReleaseNumber	Numeric[4]	Filed Traffic Flight Model (TFM = Profile) airac..
	89	71	ftfmEnvBaselineNumber	Numeric[4]	Filed Traffic Flight Model (TFM = Profile) env...
	90	72	rtfmAiracCycleReleaseNumber	Numeric[4]	Regulated Traffic Flight Model (TFM = Profile) airac..
	91	73	rtfmEnvBaselineNumber	Numeric[4]	Regulated Traffic Flight Model (TFM = Profile) env...
	92	74	ctfmAiracCycleReleaseNumber	Numeric[4]	Computed Traffic Flight Model (TFM = Profile) airac..
	93	75	ctfmEnvBaselineNumber	Numeric[4]	Computed Traffic Flight Model (TFM = Profile) env...
	94	76	lastReceivedProgressMessage	Alphabetic[3..5]	DPI, EMPTY, SIZE
SEPARATOR	95	NA	Separator	Numeric[1]	value is 0

### Example:

odralzzfdt tccSUEFFTS RPPAR nleCNSSP trFsa iio fsee llncefp stssssmree lall msrrrtt tol lall fnpo rrrft frtr etc l S  
re ci kolai aeniel l AS El l TE oqCCASUE aLeo foril ox xaooolreuaai losg x staa a eeeio B a staa lorp eeuf f f tff tfa e  
is trlnit mlrpi pii CRGa a FRsjvANVRRRpxTpb pbg i ee ttrbbiv ismptct ulclR ststn na avssBe stS s sti roerrr nm mm mm mstp  
gt uc eegeed pav saggT\_In n NOtdCeevS IPat sti gmmet ttgo ptCSCSP atu ecRS usddi e S S e t S gnpatoowa EA EA ERa  
ii sr odldasare crhhI CSnn EUppeqqQI OIHLr I n hqp Iu hu eltetde io deie e a i yys Se en Se hao inuairnir n r n r  
mneareteerirar lat t DOTE e ET enquuuVN INa d a t ditiipA ts ncoron nse v cnl tFte en ntPentlsg ttya v a v a v a  
(a d f is l pprgts otSS DRd d LI neu i i AS St l Do o ldt ST s tttal Afde eit Eiood ntPt ront CFAa ieVc Bc Bc Beit  
At t gt laair ca ncaA EADTANr ri pppLO Oc Fann al to i cizfe Fd v Ax vr ET t roSp t Pollirndi C a C a C a v c  
Ii t i i rrv r l erAM Tai YGetpmmeN Ur lit RRita a a o ei ctr A etfe eIea Sp loosSr ui FergFs y s y s y s e r  
Eo y mttta Ud NA It m S g neeqB R g a e e en tctn ch edo tfd mmFmpx loost all ongliafVluicle cle cleld

386

					5 25	55	5 5		5 5
					4 4	44	4 4		4 4
					8 8	88	8 8		8 8
EEcEH90!Cc	N6182R9o0Nrli cnss 5	5p9	X9P PINL	NN99HF	N919Nr	1 1	R C R R N N N	22C C C C	99N88C N5o 00 00 00 0 E c
IDc7IL81!M	163 810 e coasu	cd 8 X8F FIE O	88I	R68 8 e 4	4 E A E	11	A A A A	88 66M	75 55 55 5 M
KD 6H104!M	870 045 g anvr	cn 010D DIXN	00	70 0 g 3	3 A N A	33	N N N N	00 77M	08 88 88 8 P
FK 8H65!7	870 650 u oeeev	s 6 26 E G	66	76 6 u 0	0	88		66 77	3 33 33 3 T
22 5	15 2 0 eqqqe	2 32	22	72 2 0		66		22 55	Y
66 9	41 6 1 quuuq	6 46	66	76 6 1		66		66 11	
	3 0 ui i i u	0 50	00	0 0 0				00	
	8 7 i p p p i	1 61	11	1 1 7				11	
	4 p p	4 84	44	4 4				44	
		5 35	55	5 5				55	
		1 1	11	1 1				11	
		0 0	00	0 0				00	
EEcEH90!Iic	N7182R9o0P rli cnss 6	5p9	X9R R/E N Y Y	99HF	R619Nr	1 1	R C R S N N N	22C C C C	99P88ICMP6o 00 00 00 0 S c
CGd1C87!C	165 870 e coasu	cd 8 X8F FIME	88SS	V68 8 e 4	4 FIHF	22	H H H H	88 66E	56 66 66 6 I
SL 6H103!H	871 034 g anvr	cn 010P PIE X	00	70 0 g 3	3 G I	77	G G G G	00 77	97 77 77 7 Z
FL 1H60!3	871 600 u oeeev	s 6 26 RE	66	76 6 u 1	1	55		66 77	5 55 55 5 E
62 4	15 2 0 eqqqe	2 32	22	72 2 0		99		22 55	
46 2	42 6 1 quuuq	6 46	66	76 6 1		22		66 22	
	3 0 ui i i u	0 50	00	0 0 0				00	
	8 8 i p p p i	7 67	77	7 7 8				77	
	5 p p	3 83	33	3 3				33	
		0 40	00	0 0				00	
		1 1	11	1 1				11	
		2 2	22	2 2				22	
EEcEH90!Tic	N8182R9o0E rli cnss 7	5p9	X9F FIS L	NN99SIT	N919Nr	1 1	R C R R Y Y Y	22C C C C	99N88T CE7o 00 00 00 0 S c
IGc7IC87!A	167 870 e coasu	cd 8 X8P PIE O	88I	V68 8 e 4	4 R N R	44	N N N N	88 66A	47 77 77 7 I
KP 3H205!H	872 054 g anvr	cn 010L LVRN	00	70 0 g 3	3 N L N	11	L L L L	00 77P	86 66 66 6 Z
FF SH60!6	872 605 u oeeev	s 6 26 E G	66	76 6 u 2	2	33		66 77	7 77 77 7 E
62 c	15 2 0 eqqqe	2 32	22	72 2 0		11		22 55	
46 2	43 6 1 quuuq	6 46	66	76 6 1		88		66 33	
	3 0 ui i i u	0 50	00	0 0 0				00	
	8 9 i p p p i	7 67	77	7 7 9				77	
	6 p p	5 85	55	5 5				55	
		0 50	00	0 0				00	
		5 5	55	5 5				55	
		4 4	44	4 4				44	
EEcEH91!Iic	N9182R91oR rli cnss 8	5p9	X9N N H N Y Y	99TT	N919Nr	1 1	R D R S Y Y Y	22D D D D	99P88ICAT8o 00 00 00 0 S c
CKc7C82!C	169 820 e coasu	cd 8 X8O OTE E	88AA	R68 8 e 4	4 R E R	55	E E E E	88 66E	38 88 88 8 I
SC 5SC00!H	873 005 g anvr	cn 010N NIA X	00	70 0 g 3	3 P P P	55	P P P P	00 77	75 55 55 5 Z
FH 8H60!4	873 600 u oeeev	s 6 26 IDE	66	76 6 u 3	3	00		66 77	9 99 99 9 E
62 3	15 2 0 eqqqe	2 32	22	72 2 0		44		22 55	
66 c	44 6 1 quuuq	6 46	66	76 6 1		44		66 44	
	3 1 ui i i u	1 51	11	1 1 1				11	
	8 0 i p p p i	2 62	22	2 2 0				22	
	7 p p	0 80	00	0 0				00	
		0 60	00	0 0				00	
		4 4	44	4 4				44	
		7 7	77	7 7				77	
EEcEH90!Ez	Yc1183R9o0T rli cnss 9	5p9	X9P PIAL	NN99A	R619Nr	1 1	S D S R N N N	22E E E E	99N88Z CR9o 00 00 00 0 c
IDc1IL88!IZ	0161 880 e coasu	cd 8 X8F FIE O	88AA	V68 8 e 4	4 A L A	66	R R R R	88 66Z	29 99 99 9
KD 3H00!IZ	874 004 g anvr	cn 010D DIAN	00	70 0 g 3	3 M A M	88	R R R R	00 77Z	65 55 55 5
FK 1H60!C	874 605 u oeeev	s 6 26 P G	66	76 6 u 4	4	77		66 77	1 11 11 1
72 I	15 2 0 eqqqe	2 32	22	72 2 0		77		22 55	
86 4	45 6 1 quuuq	6 46	66	76 6 1		00		66 55	
6	3 1 ui i i u	0 50	00	0 0 1				00	
	8 1 i p p p i	8 68	88	8 8 1				88	
	8 p p	0 80	00	0 0				00	
		0 70	00	0 0				00	
		2 2	22	2 2				22	
		5 5	55	5 5				55	

## 5.2.21 exp - Expand file

Expand	
extension	exp
origin	traffic demand data
separator	blank
sort	no
comment	input for SAAM modeling (assignment ...)

#	Field	Type	Size	Comment
1	origin	char		ICAO code (airport)
2	destination	char	4	ICAO code (airport)
3	not used	num	2	"oo"
4	aircraft type	char	4	
5	RFL	num	3	Warning: generally the value is calculated based on the maximum flight level reached.
6	zone origin	char	4	ICAO code (could be the same as airport)
7	zone destin	char	4	ICAO code (could be the same as airport)
8	flight ID	num		SAMAD ID or SAAM ID or ... (must be uniq)
9	date departure	num	6	YYMMDD
10	time departure	num	4	HHMM
11	time arrival	num	4	NOT USED = "!!!!"
12	callsign	char	till 7	
13	company	char	3	generally the 3 first letters of the callsign

### Example (10 flights):

```
BGSF EKCH 00 B767 290 BGSF EKCH 367836 980626 1445 !!!! SAS292 SAS
BIKF EDDF 00 B757 370 BIKF EDDF 328474 980626 0745 !!!! ICE520 ICE
BIKF EDDK 00 CL60 290 BIKF EDDK 371896 980626 1516 !!!! GAF208 GAF
BIKF EDDK 00 B73S 270 BIKF EDDK 337460 980626 1740 !!!! ICE774 ICE
BIKF EDDL 00 B757 370 BIKF EDDL 346578 980626 0105 !!!! LTU1929 LTU
BIKF EDDM 00 B757 350 BIKF EDDM 351193 980626 0145 !!!! CMM759 CMM
BIKF EFHK 00 B73S 330 BIKF EFHK 328169 980626 0730 !!!! ICE342 ICE
BIKF EGLL 00 B757 290 BIKF EGLL 364422 980626 0750 !!!! ICE450 ICE
BIKF EGPF 00 B73S 290 BIKF EGPF 364457 980626 1200 !!!! ICE430 ICE
BIKF EGUN 00 C130 210 BIKF EGSS 366363 980626 0800 !!!! BLIGH49 ZZZ
```

## 5.2.22 flc2 - Flight Level Constraints

Flight level constraint file, flc2, is included in SAAM dataset file, that can be down-loaded from [DDR](#). It uses regular expression to mimic flight level restrictions found in Route Availability Document (RAD). It contains also flight level restrictions needed for SAAM Profile process to reflect reality.

All flight level constraints are grouped by country, using ICAO country code. Each constraint has an explanation of it's origin, either a translated RAD restriction, observed by CFMU or needed by SAAM.

Go to [Modifying the Assignment & Profile Parameters/Flight Level Constraints](#) to read more about the file format.



### 5.2.23 fld - Flow Demand file

Flow Demand				
extension	Fld			
origin	SAAM processing or CFMU data (via PRISME)			
separator	Blank (i.e. Space)			
sort	No			
comment	Used as an input in 3D density, forecast, transform fld to so6, etc ...			

#	Field	Type	Size	Comment
1	origin of flight	char	4	ICAO code
2	destination of flight	char	4	ICAO code
3	Number of flights	Num		
4	Lat 1	num		in minute decimal
6	Long 1	num		in minute decimal
7	Lat 2	num		in minute decimal
8	Long 2	num		in minute decimal

#### Example:

```
LEBL LEMD 21525 2477.8167 124.7000 2428.3333 -213.6500
LEMD LEBL 21405 2428.3333 -213.6500 2477.8167 124.7000
LIRF LIML 12728 2508.6667 735.1500 2727.0167 556.7667
LIML LIRF 12685 2727.0167 556.7667 2508.6667 735.1500
LFPO LFMN 12451 2923.4000 142.7667 2619.9167 432.9000
LFMN LFPO 12419 2619.9167 432.9000 2923.4000 142.7667
LFPO LFBO 12263 2923.4000 142.7667 2618.1000 82.0667
LFBO LFPO 12244 2618.1000 82.0667 2923.4000 142.7667
LEBL LEPA 11728 2477.8167 124.7000 2373.1000 164.3333
LEPA LEBL 11427 2373.1000 164.3333 2477.8167 124.7000
LFPO LFML 10921 2923.4000 142.7667 2606.2000 312.9000
LFML LFPO 10919 2606.2000 312.9000 2923.4000 142.7667
```

### 5.2.24 flf - Flight Info file

Flight inFo				
extension	flf			
origin	cfmu			
separator	blank			
sort	no			
comment	input for SAAM traffic filtering and trajectories matching			

#	Field	Type	Size	Comment
1	Flight SAAM ID	num		uniq for the day
2	Flight SAMAD ID	num		uniq for ever
3	TACT ID	num		uniq for ever, CFMU ID
4	SSR_CODE	char		
5	REGISTRATION	char		
6	Planned Date departure	num	6	YYMMDD padded with o's
7	Planned Time departure	num	4	HHMM padded with o's

8	ATFM DELAY	num		
9	REROUTING STATE	char		
10	most pen reg	char		if no regulation 'X' is present
11	type	char		letters indicating type of flight (military ?)
12	equipment	char		letters indicating equipment of flight
13	ICAO equipment	char		
14	COM equipment	char		
15	NAV equipment	char		
16	SSR equipment	char		
17	SURVIVAL equip.	char		
18	PERSONS ON BOARD	num		0 means no information
19	top FL	num		
20	max RFL	num		
21	FLT PLN SOURCE	char		

### 5.2.25 frp - Free Route Points file

Free Route Points	
extension	frp
origin	saam
separator	blank or tabulation
sort	no
comment	used during SAAM assignment to generate Free Route segments

#	Field	Type	Size	Comment
1	Free Route Area name	char		name used in the associated sls file
2	Point type	char		Entry or Exit or Both, or Intermediate point for military areas. Possible values for Entry are E/D, possible values for Exit are X/A value for Intermediate point is I
3	Point name	char		if not an Intermediate Point, must be present in ase network files
4	Latitude	N   S d d m m s s		degrees, minutes, seconds. Used only for E/X/I points and optionally for E/X points
5	Longitude	E   W d d m m s s		degrees, minutes, seconds. Used only for E/X/I points and optionally for E/X points
4.. N	Airport name	char		Airport restriction : Free route to or from the point allowed only with this Arrival/Departure set of airports. Used only for A/D points and optionally. Name can contain '.' character to match any A-Z character.

#### Example:

```

LISBOA EX RETEN
LISBOA EX ARMED
LISBOA EX BANAL
LISBOA EX DETOX
LISBOA EX ERPES
LISBOA EX GUNTI

```

```
LISBOA EX KOMUT
LISBOA EX LUTAK
LISBOA EX MANOX
LISBOA EX PETEK
LISBOA D CP
LISBOA A LAR
LISBOA AD POR
LISBOA A TROIA LPPT LPCS
LISBOA AD GAIOS LPPT
LISBOA D DIRMA LP..
LISBOA D LULAS LPPR
LISBOA D MANIK
LISBOA D PESUL
LISBOA D IBERO
LISBOA A XAPIM
LISBOA A GOBIX
LISBOA D VFA
LISBOA AD SNT
SWEDENNORTH EX GIKAV
SWEDENNORTH EX SOLKA
SWEDENNORTH X TIGBA
SWEDENNORTH X OBIVO
SWEDENNORTH EX KELVI
SWEDENNORTH EX SUVAR
SWEDENNORTH EX BUGAX
SWEDENNORTH EX VEKOX
SWEDENNORTH EX ODNEL
SWEDENNORTH EX ENITI
SWEDENNORTH EX ASNIK
SWEDENNORTH EX AKAVU
SWEDENNORTH E PIPIV
SWEDENNORTH EX UMLAX
SWEDENNORTH EX XAXAX
SWEDENNORTH E RUSUT
SWEDENNORTH E RADUP
SWEDENNORTH E ARTAB
SWEDENNORTH X KAPOG
SWEDENNORTH X EVSIN
SWEDENNORTH X ESEBI
SWEDENNORTH EX VEVIG
SWEDENNORTH EX BAKLA
SWEDENNORTH EX TOGMI
SWEDENNORTH A SLU01
SWEDENNORTH D BESLA
SWEDENNORTH D VERAG
SWEDENNORTH D RISEM
SWEDENNORTH A UME09
SWEDENNORTH A UME03
SWEDENNORTH D TINDO
SWEDENNORTH D DEDIT
SWEDENNORTH D SOPLI
SWEDENNORTH AD SUN
SWEDENNORTH AD OSS
SWEDENNORTH AD SKE ESNO ESNU
SWEDENNORTH AD KRA
SWEDENNORTH AD ESNLs
Irland EX MIMKU
Irland EX IBROD
Irland EX GOMUP
Irland I AMPED N513400 W0111300
```

```

Irland I UNLID N511233 W0104329
Irland I LINRA N513447 W0100156
LS X CROSS N470902 E0093614
LO E CROSS N470902 E0093614

```

## 5.2.26 gar - Gasel Airblock file

Airblock	
extension	gar
origin	Gasel, SAAM
separator	;
sort	no sort
comment	Describe 2D definition of airblocks see Converting SAAM format to/from Gasel format

First line of the file				
#	Field	Type	Size	Comment
1	nb	num	1	number of airblocks defined

Header				
#	Field	Type	Size	Comment
1	tag	char	1	must be "A"
2	name of airblock	char	~	ID (max 24 characters for SAAM usage). Must be the same as the one found in associated ".gsl" file
3	number of items	num	~	number of items (vertices) composing the airblock (indicate the number of body lines) the first item being repeated at the end of the body

Body, made with Header "number of items" lines				
#	Field	Type	Size	Comment
1	tag	char	1	must be "P"
2	latitude	num	~	in degrees decimals
3	longitude	num	~	in degrees decimals

### Example: 2 airblocks

```

2
A;000EG;26
P;57;-15
P;57;-14.015
P;57;-13
P;57;-10
P;56.66666666666667;-10
P;56.5;-10
P;56.25722222222222;-10
P;56;-10
P;55.5;-10
P;55;-10
P;54.6;-10
P;54.56666666666667;-10
P;54.5;-10.66666666666667
P;54.36666666666667;-12
P;54.25333333333333;-13
P;54.18333333333333;-13.5

```

```

P;54;-15
P;54.25;-15
P;54.5;-15
P;55;-15
P;55.25;-15
P;55.5;-15
P;56;-15
P;56.25;-15
P;56.5;-15
P;57;-15
A;001BI;14
P;64.63333333333333;-22.966666666666667
P;64.78333333333333;-22.883333333333333
P;64.71666666666667;-22.466666666666667
P;64.7;-22.116666666666667
P;64.66666666666667;-21.3
P;64.28333333333333;-20.95
P;63.86666666666667;-21.283333333333333
P;63.71666666666667;-21.216666666666667
P;63.47611111111111;-21.646111111111111
P;63.32027777777778;-22.614444444444444
P;63.55833333333333;-23.77
P;64.215;-24.048611111111111
P;64.4975;-23.600555555555556
P;64.63333333333333;-22.966666666666667

```

### 5.2.27 gr1 - Sector Load Input Chart Data file

Sector Load input chart data format	
extension	gr1
origin	SAAM sector load
separator	horizontal tabulation (displayed as → in the example below)
sort	
comment	<p>ascii file describing the data to be displayed in charts, used by <a href="#">SAAM chart</a> and combine curve.</p> <p>A gr1 file may contains several charts data. Each chart is made of:            "Chart Header line1" followed by "Chart Header Line 2" followed by            "Chart Body" followed by "Chart Body End"</p> <p>Note: the number of columns depends on the number of series in the graphs.</p>

Global GR1 File Header line				
#	Field	Type	Size	Comment
1	numchart	num		number of charts described in the file (below)

Chart Header line 1 (TAB SEPARATED)				
#	Field	Type	Size	Comment
1		char	0	empty field
2	chart name	char		name of the chart (in general the name of the airspace)
3	x_name	char		name of x axe (in general "Time")
4	y_name	char		name of y axe (in general "Number of flights")

Chart Header line 2 (TAB SEPARATED)				
#	Field	Type	Size	Comment
1		char	0	empty field
2		char	0,1	one space or empty field (ignored)
3	curve name	char		name of first curve (for example: "SHER")

	1			
4	curve name	char		name of first curve (for example: "Inst. Count")
2				
...	...	char		...
n	curve name	char		name of curve n
n				

**Chart Body (TAB SEPARATED) is made in general of x lines to cover 24hours by step of (24\*60)/x minutes**

#	Field	Type	Size	Comment
1	chart name	char		name of the chart
2	time	int	4	expressed HHMM padded with 0
3	value curve 1	int, float		value for curve 1 (can have decimals)
4	value curve 2	int, float		value for curve 2 (can have decimals)
...	...	int, float		...
n	value curve n	int, float		value for curve n (can have decimals)

#### Chart Body End

#	Field	Type	Size	Comment
1	empty	char	0	empty line (mandatory)

### Example (2 charts with time step of 6 hours)

```

2
→ENGMTMA→Time→Nbr of flights
→→SHER→INST
ENGMTMA→0000→0→0
ENGMTMA→0600→0→0
ENGMTMA→1200→1→1
ENGMTMA→1800→1→1

→SLEMMCCO→Time→Nbr of flights
→ →SHER→INST
SLEMMCCO→0000→1→1
SLEMMCCO→0600→1→1
SLEMMCCO→1200→1→1
SLEMMCCO→1800→1→1

```

## 5.2.28 gsl - Gasel Sector file

Sector	
extension	gsl
origin	Gasel, SAAM
separator	;
sort	no sort
comment	Describe how elementary sectors are built from airblocks

#### First line of the file

#	Field	Type	Size	Comment
1	nb	num	1	number of elementary sectors defined

#### Header

#	Field	Type	Size	Comment
1	tag	char	1	must be "S"
2	name of sector	char	~	ID, used in other files, like spc or cfg (max 19 characters for SAAM usage)
3	comment	char	~	free text describing the sector, can be empty
4	number of items	num	~	number of items composing the sector (indicate the number of body lines)

**Body, made with Header "number of items" lines**

#	Field	Type	Size	Comment
1	tag	char	1	must be "A"
2	name of airblock	char	~	ID, may be used several times in the file (max 24 characters for SAAM usage)
3	construction op	char	1	must be "+"
4	bottom level	num	~	in FL
5	top level	num	~	in FL

**Example: 4 sectors**

```

4
S;BIRDES;REYKJAVIK EAST;4
A;003BI;+;85;999
A;100BI;+;55;999
A;105BI;+;195;999
A;110BI;+;55;999
S;BIRDFAXI;_;1
A;001BI;+;55;245
S;BIRDFIS;_;7
A;001BI;+;0;55
A;002BI;+;0;55
A;004BI;+;0;55
A;100BI;+;0;55
A;104BI;+;0;55
A;107BI;+;0;55
A;110BI;+;0;55
S;BIRDNO;BIRD NORTH SECTOR;2
A;103BI;+;195;999
A;104BI;+;55;999

```

## 5.2.29 interieur

Interieur.txt				
extension:	interieur.txt			
origin:	icr.exe			
separator:	blank			
sort:	by window (field 3), then by seg sequence (field 11)			
comment:	it is the result of intersection between 2D windows (.are file) and segments (name1 name2 lat1 lon1 lat2 lon2) Describes segment lying inside the windows			
#	Field	Type	Size	Comment
1	segment name	char		Built with point name separated with " _ "
2	area name	char		is field 15 of ".are" file. "ZZ" means no window
3	area index	num		index starts at 1

4	point name seg origin	char		
5	latitude seg origin	float		in minute decimal
6	longitude seg origin	float		in minute decimal
7	point name seg end	char		
8	latitude seg end	float		in minute decimal
9	longitude seg end	float		in minute decimal
10	segment distance	float		in nm decimal, is the total length of the segment from origin to end
11	seg sequence	num		"-1" means entire segment in the window, else segment sequence (start at 1)
12	intersec begin lat	float		where seg enters window, set to origin if seg origin is in window
13	intersec begin lon	float		where seg enters window, set to origin if seg origin is in window
14	distance begin	float		distance from origin to intersection begin, 0 if seg origin is in window
15	intersec end lat	float		where seg quits window, set to end if seg end is in window
16	intersec end lon	float		where seg quits window, set to end if seg end is in window
17	distance end	float		distance from origin to intersection end, total length if seg end is in window

**Example:**

```
BARIT_KOMAR ZLDZOCD1 1 BARIT 2595.756383 1071.140600 KOMAR 2697.775117 1009.029317 111
KOMAR_BARIT ZLDZOCD1 1 KOMAR 2697.775117 1009.029317 BARIT 2595.756383 1071.140600 111
GILUK_BOSNA ZLDZOCD1 1 GILUK 2632.000000 1088.000000 BOSNA 2662.385817 1065.994550 34.
KEB_BOSNA ZLDZOCD1 1 KEB 2636.000000 1107.000000 BOSNA 2662.385817 1065.994550 39.5191
BOSNA_KOMAR ZLDZOCD1 1 BOSNA 2662.385817 1065.994550 KOMAR 2697.775117 1009.029317 53.
```

**5.2.30 jse - Jet Stream Effect file**

Jet Stream Effect				
extension	jse			
origin	produced by SAAM jetStreamAnalyser, used by jetstream (normaly launch immediately after profile processing)			
separator	blank			
sort	no sort			
comment	each line contains a city pair and the associated time shift information for the arrival			
#	Field	Type	Size	Comment
1	origin	char		ICAO code
2	destination	char		ICAO code
3	time shift at the arrival	int		unit is second

**Example:**

```
KJFK EGLL 4120
KJFK LEMD 120
```



EDDF KJFK 0

### 5.2.31 lay - Layer file

Layer				
extension:	lay			
origin:	SAAM			
separator:	semi column ";"			
sort:	No sort			
comment:	Describe an Airspace Design project which will be linked to most of SAAM data files			

#	Field	Type	Size	Comment
1	Version	num		indicates formatting version of this file, currently = 1
2	Project ID	num		unique identifier of the project, used as a key to other SAAM files, could be derived from current system date and time
3	Scenario name	char		name of a scenario. A slash "/" introduce hierarchy between projects
4	Effective date Beg.	char	10	starting date expressed dd/mm/yyyy from which item can be operated, special case 01/01/1900 means effective date is not known and/or then item can always be operated.
5	Effective date End	char	10	ending date expressed dd/mm/yyyy from which item can be operated, special case 31/12/9999 means effective date is not known and/or item can always be operated.
6	Type	char		could be used to indicates the type of the project "NETWORK", "AIRSPACE", "MILITARY", "CONTINGENCY", "RAD"... could be empty.
7	Author	char		name of the responsible of the project
8	Description	char		brief description of the project. Could be empty.
9	Project date	char	10	date when the project started to be studied (input) in SAAM. Format: dd/mm/yyyy

**Example (do not correspond to any official project):**

```
1;220120091649;FAB/FABEC/German Swiss Interface;01/04/2010;31/12/9999;BOTH;Robert;inve
```

### 5.2.32 lop - List of Options file

List of Options				
extension:	lop			
origin:	SAAM			
separator:	blank			
sort:	NO, but CDR filter requires LOP to be sorted by original flight ID (field 2) then by flight option number (field 10)			
comment:	Describe 2D/3D/4D route option features Input and output for SOP, query, CDR management. Created by Route Option Generator.			

#	Field	Type	Size	Comment
1	flight ID	num		* is original flight ID for flight having a ref. routing, OR * >=1000000000 for flight having a new routing option (generally longer than ref. routing)
2	original flight ID	num		original/father flight, as found in expand file

3	validation	num		1 (default) means option is valid (possibly visually checked by expert) else 0
4	choice	num		0 (default) means option was not chosen by Optimiser else 1
5	extension cost	num		route length cost of the option expressed in nm
6	global cost	num		length cost OR total cost in EURO (=length cost * N EURO/NM + CRCO cost)
7	departure	char	4	Airport ICAO code
8	arrival	char	4	Airport ICAO code
9	flow ID	num		ID of flow (set of flights having same airport or zone arrival departure), see equi file
10	flight option number	num		starts at 1 for every flight, then goes to 2 if the flight have a routing option...
11	flow demand	num	4	for MMFA, fixed to 1/14 hour (= 0.0714286)
12	flow selected	num		for MMFA, generally set to 0
13	cost delay	num		for MMFA, generally set to 1800

**Example (4 original/father flights, 2 of them have 2 additional route options):**

```
615768 615768 1 0 1221.3 1221.3 LDDU EIDW 27061 1 0.0714286 0 1800
644947 644947 1 0 1221.3 1221.3 LDDU EIDW 27061 1 0.0714286 0 1800
34081 34081 1 0 763.191 763.191 LDDU EDDL 34081 1 0.0714286 0 1800
1000000000 34081 1 0 772.942 772.942 LDDU EDDL 34081 2 0.0714286 0 1800
1000000001 34081 1 0 782.873 782.873 LDDU EDDL 34081 3 0.0714286 0 1800
616502 616502 1 0 763.191 763.191 LDDU EDDL 34081 1 0.0714286 0 1800
1000000002 616502 1 0 772.942 772.942 LDDU EDDL 34081 2 0.0714286 0 1800
1000000003 616502 1 0 782.873 782.873 LDDU EDDL 34081 3 0.0714286 0 1800
```

### 5.2.33 lox - List of Options Extended file

List of Options eXtended	
extension:	lox
origin:	SAAM
separator:	blank
sort:	NO, but CDR filter requires LOX to be sorted by original flight ID (field 2) then by flight option number (field 10)
comment:	Describe 2D/3D/4D route option features Input and output for query, optimizer. Created by Assignment.

#	Field	Type	Size	Comment
1	flight ID	num		* is original flight ID for flight having a ref. routing, OR * >=1000000000 for flight having a new routing option (generally longer than ref. routing)
2	original flight ID	num		original/father flight, as found in expand file
3	validation	num		1 (default) means option is valid (possibly visually checked by expert) else 0
4	choice	num		0 (default) means option was not chosen by Optimiser else 1
5	status	num	4	status of the option (2D/3D/4D) : x1x2x3x4 with xi=0/1, x1=2D option, x2=RFL option, x3=FL constraint option, x4=4D option
6	global cost	num		length cost OR time cost or total cost in EURO
7	departure	char	4	Airport ICAO code
8	arrival	char	4	Airport ICAO code

9	flow ID	num		ID of flow (set of flights having same airport or zone arrival departure), see equi file
10	flight option number	num		starts at 1 for every flight, then goes to 2 if the flight have a routing option...
11	original departure time	num	4	HHMM
12	departure delay	num		in minutes, can be negative if departure is before original departure time
13	original RFL	num	3	
14	delta RFL	num		
15	original flight duration	num		in seconds
16	delta flight duration	num		in seconds
17	route length	num		distance of the route flown between origin & destination expressed in NM

### Example (2 original/father flights, 1 of them have 16 options):

```

108471347 108471347 1 0 0000 7873 LEIB EGCC 108459205 1 0950 0 400 0 7873 0 881.702
108471420 108471420 1 0 0000 7173 EIDW EPSC 108471420 1 1005 0 350 0 7173 0 752.778
1000015654 108471420 1 0 0010 7157 EIDW EPSC 108471420 2 1005 0 350 0 7173 -16 752.778
1000015655 108471420 1 0 0100 7197 EIDW EPSC 108471420 3 1005 0 350 20 7173 24 752.778
1000015656 108471420 1 0 0110 7169 EIDW EPSC 108471420 4 1005 0 350 20 7173 -4 752.778
1000015657 108471420 1 0 0001 7173 EIDW EPSC 108471420 5 1005 -20 350 0 7173 0 752.778
1000015658 108471420 1 0 0001 7173 EIDW EPSC 108471420 6 1005 20 350 0 7173 0 752.778
1000015659 108471420 1 0 0001 7173 EIDW EPSC 108471420 7 1005 40 350 0 7173 0 752.778
1000015660 108471420 1 0 0011 7157 EIDW EPSC 108471420 8 1005 -20 350 0 7173 -16 752.7
1000015661 108471420 1 0 0011 7157 EIDW EPSC 108471420 9 1005 20 350 0 7173 -16 752.7
1000015662 108471420 1 0 0011 7157 EIDW EPSC 108471420 10 1005 40 350 0 7173 -16 752.7
1000015663 108471420 1 0 0101 7197 EIDW EPSC 108471420 11 1005 -20 350 20 7173 24 752.
1000015664 108471420 1 0 0101 7197 EIDW EPSC 108471420 12 1005 20 350 20 7173 24 752.7
1000015665 108471420 1 0 0101 7197 EIDW EPSC 108471420 13 1005 40 350 20 7173 24 752.7
1000015666 108471420 1 0 0111 7169 EIDW EPSC 108471420 14 1005 -20 350 20 7173 -4 752.
1000015667 108471420 1 0 0111 7169 EIDW EPSC 108471420 15 1005 20 350 20 7173 -4 752.7
1000015668 108471420 1 0 0111 7169 EIDW EPSC 108471420 16 1005 40 350 20 7173 -4 752.7
1000015669 108471420 1 0 1000 7260 EIDW EPSC 108471420 17 1005 0 350 0 7173 87 752.778

```

### 5.2.34 lsr - List of SOP Runs file

List of SOP Runs				
extension	lsr			
origin	SAAM			
separator	blank			
sort	NO			
comment	input and output for SOP, query ...			
#	Field	Type	Size	Comment
1	run ID	num		number identifying the run, used in field #4 of lop file
2	mono/multi	num		mono-assignment or multi-assignment
3	type	num		MMFA or SOP or ...
4	percentage optim.	num		percentage of optimisation
5	time	num		time of run
6	date	num		date of run
7	author	char		name of person who made the run

9	study	char		name of study
	fichier option			
	fichier t5			
	fichier capa			
	default capa			
	coef			
8	comment	num		free comment

### 5.2.35 mot - Military Opening Time file

Airspace Activation Time	
extension:	mot
origin:	manual input. Editable with the <a href="#">Airspace Activation Editor</a> as well.
separator:	blank
sort:	no
comment:	input for SAAM processing: Route choice Based on Military Opening Time and Assignment with profile restrictions (version 3)

#	Field	Type	Size	Comment
1	version number	num		1 (only time) or 2 (time and level) or 3 (time and type)
2	military zone name	char		must be the name appearing in first column of SLS file (normal name)
3	begin time	num	4	HHMM padded with zeros
4	end time	num	4	HHMM padded with zeros
5	lower FL	num	3	Flight Level with 0 (for instance 055) this field exists if version is 2
6	upper FL	num	3	Flight Level with 0 (for instance 055) this field exists if version is 2
5	type of opening	char	2	EX (=opening time from Entry to Exit) or E (= opening time at Entry, no restriction at Exit) this field exists if version is 3

#### Example (4 military zones) with the first 2 format versions:

```

1 MIL1 1000 1200
1 MIL1 1400 1700
1 MIL2 0900 1300
1 MIL3 1100 1300
1 MIL4 0900 1800

2 MIL1 1000 1200 055 245
2 MIL1 1400 1700 085 285
2 MIL2 0900 1300 000 600
2 MIL3 1100 1300 125 245
2 MIL4 0900 1800 000 285

```

#### Example (3 Free route areas) with the format version 3:

```

3 ED 2200 0500 EX
3 LF 2300 0600 EX
3 LI 2200 0500 E

```

**Notes:**

A time period (between begin time and end time) indicates Airspace activity (closed for Civil traffic in case of military airspace). For assignment, it indicates airspace opening for GAT.

A zone can have several different time period (see example with MIL1 zone).

If a zone does not appear in MOT file it is considered open for civil traffic in [Route Choice Based On Military](#) context.

In the last given examples, the zone is activated during night and the activity time window extends around midnight. To specify an activity time ending at midnight, you shall write 2400.

**5.2.36 Old SAAM are file**

Newmaxo ascii region file	
extension	are
origin:	newmaxo, SAAM, CAPAN
separator	blank
sort	Body is sorted by point sequence (for SAAM it should be clockwise)
comment:	has a header and a body. First the header( one line) followed by the body (n lines).

Header				
#	Field	Type	Size	Comment
1	nb_point	num	~	contains the number of line of the following body
2	latitude	num	~	in minutes (decimal for SAAM), location of the label
3	longitude	num	~	in minutes (decimal for SAAM), location of the label
4	flights	num	~	number of flights slot of info
5	low_level	num	~	low level of the volume in flight level (FL)
6	high_level	num	~	high level of the volume in flight level (FL)
7	surface	num	~	surface of the volume, not important can be 0
8	sector_num	num	~	number of sector in the volume, not important can be 0
9	flight time	num	~	flight time, not important can be 0
10	traffic density	num	~	traffic density, not important can be 0
11	x mileage	num	~	extra mileage, not important can be 0
12	rte extens.	num	~	Store TopLev/BotLev/Trans/Light/Top codes (SAAM code)
13	value 1	num	~	Store the color+trans of the Airblock (SAAM code)
14	value 2	num	~	Store the sign of the airblock (CAPAN), 0=positif, 1=negatif
15	name	char	~	name or code of the volume (max 12 char for newmaxo)

Body (the polygon must be closed: first point = last point)				
#	Field	Type	Size	Comment
1	latitude	num	~	in minutes (in minutes decimal for SAAM)

2	longitude	num	~	in minutes (in minutes decimal for SAAM)
---	-----------	-----	---	--

**Comment: the SAAM codes are:**

=0 means the volume properties is set by the one found in TDV line

=1 means TopLev or BotLev set by field #5 and #6 or value is ON

=2 means value is OFF

!=0 means for color only, the value of the color

Color has 4 components varying between 0 to 255= $\text{trans} \times 255^3 + \text{red} \times 255^2 + \text{green} \times 255 + \text{blue}$

The blue component of the color, if setup, is never equals to ZERO.

field #12 =  $\text{TopLev} \times 256 + \text{BotLev} \times 64 + \text{trans} \times 16 + \text{Light} \times 4 + \text{top}$

**Example:**

```
14 2799 925 0 0 660 0 0 0 0 0 0 0 0 607LJ
2799 925
2792 932
2797 943
2785 954
2780 942
2774 947
2781 957
2784 964
2784 975
2784 977
2790 974
2794 982
2790 992
2799 925
```

**5.2.37 ord - Sector Order file**

sector order				
extension	ord			
origin	User manual input			
separator	N/A			
sort	sort the lines (containing sector names) according to the order you want the sector loads to be displayed			
comment	each line contains the name of a sector for which the sector load has to be computed			
#	Field	Type	Size	Comment
1	sector name	char	26	the same one as the one found in field 2 of t5

**Example:**

```
LFEPPLOW
EDDYLN0
```

**5.2.38 pin zin - Profile Input file**

profin	
extension	pin
:	
origin:	SAAM kernel
separator	blank
:	
sort:	no
comment	input of SAAM profile processing. Each flight has a header (one line) and a body (n lines).

:	
	Note that the distances need not be linear distances and should therefore not be recalculated from coordinates.

HEADER				
#	Field	Type	Size	Comment
1	flight identifier	num		
2	origin of flight	char	4	ICAO code
3	destination of flight	char	4	ICAO code
4	aircraft type	char	4	
5	route length	float		total length of the route
6	departure time	num		expressed in minute from midnight
7	RFL	num		Optional: if RFL=0 or RFL outbound then Optimum Cruise FL is taken
8	body lines number	num		number of following body lines
9	callsign	char	<=7	
10	departure date	num	6	YYMMDD padded with o's

BODY				
#	Field	Type	Size	Comment
1	flight identifier	num		same as the header one
2	point name	char	3 or 5	
3	length	float		length from origin to this point
4	parity	num	1	0=NO, 1=ODD, 2=EVEN, 3=ODD_LOW, 4=EVEN_LOW, 5=ODD_HIGH, 6=EVEN_HIGH (from this point to next one)
5	level constraint	num	1 to 4	NEVER starts with oxx 999=no constraint, +nnn=not above FLnnn in climbing phase, -nnn=not above FLnnn in descending phase, if the constraint is set on the first point of the route (departure, with length=0) it means not above FLnnn in cruising phase of the flight.
6	latitude	float		expressed in minute of angle (with decimals)
7	longitude	float		expressed in minute of angle (with decimals)

**Example (3 flights):**

```

110 DAAG LEBL B727 331 900 0 6 DAH2012 970905
110 DAAG 0 2 999 2202 193
110 ALR 3 1 999 2200 190
110 MHN 200 2 999 2391 253
110 TASOS 298 2 999 2456 158
110 QUV 330 2 999 2477 125
110 LEBL 331 0 999 2477 124
111 DAAG LEPA B727 234 1090 330 5 DAH2000 970905
111 DAAG 0 2 999 2202 193
111 ALR 3 2 999 2200 190
111 IBZ 160 2 999 2334 88
111 MJV 227 2 999 2366 165

```

```

111 LEPA 234 0 999 2373 163
239 DTTA LICJ AT42 271 490 0 6 AZA873 970905
239 DTTA 0 1 999 2211 614
239 TUC 0 1 999 2211 614
239 MON 70 1 999 2143 637
239 TRP 229 1 999 2273 750
239 PAL 261 2 999 2281 790
239 LICJ 271 0 999 2290 786

```

### 5.2.39 pl6 - PointLine6 file

PointLine6				
extension:	pl6			
origin:	produced by so6ToPl, used by plToSo6. Different from the one provided by Eatchip Database (field #14 is missing)			
separator:	blank			
sort:	Yes on field #1 (id) then on Field #3 (seq)			
comment:	same format as so6 but point by point (files are smaller)			

#	Field	Type	Size	Comment
1	ID	num		flight ID
2	break	num		same number= no break, new number=break
3	sequence	num		point sequence, start at 1 for the first point
4	point name	char		
5	level	num		Unit is Flight Level
6	time	num	6	Format is HHMMSS with leading zeros
7	date	num	6	Format is YYMMDD with leading zeros
8	origin	char		airport where the flight is taking off
9	destination	char		airport where the flight is landing
10	aircraft type	char		
11	callsign	char		
12	status	num	1	to next point: 0=climb, 1=descent, 2=cruise
13	parity	num	1	to next point: 0=NO, 1=ODD, 2=EVEN, 3=ODD_LOW, 4=EVEN_LOW, 5=ODD_HIGH, 6=EVEN_HIGH
14	length ?	num	1	ALWAYS ZERO (=0)
15	latitude	float		in minute decimal
16	longitude	float		in minute decimal

#### Example (1 flight):

```

23 1 1 BIKF 0 072000 010626 BIKF ELLX B757 ICE614 0 1 0 3839 -1356
23 1 2 $aksP 390 074230 010626 BIKF ELLX B757 ICE614 0 1 0 3782.48 -1066.26
23 1 3 B 390 081604 010626 BIKF ELLX B757 ICE614 2 1 0 3660 -600
23 1 4 *BORM 390 085348 010626 BIKF ELLX B757 ICE614 2 1 0 3469 -193
23 1 5 ADN 390 085938 010626 BIKF ELLX B757 ICE614 2 1 0 3438 -136
23 1 6 SKATE 390 092949 010626 BIKF ELLX B757 ICE614 2 1 0 3300 184
23 1 7 SAMON 390 093810 010626 BIKF ELLX B757 ICE614 2 1 0 3243 227
23 1 8 SPY 390 095123 010626 BIKF ELLX B757 ICE614 2 1 0 3152 291
23 1 9 $aksQ 390 100140 010626 BIKF ELLX B757 ICE614 2 1 0 3078.15 324.359
23 1 10 LNO 244 100802 010626 BIKF ELLX B757 ICE614 1 1 0 3035 343
23 1 11 LARED 208 100950 010626 BIKF ELLX B757 ICE614 1 1 0 3024 349
23 1 12 DIK 68 101539 010626 BIKF ELLX B757 ICE614 1 1 0 2992 368
23 1 13 ELLX 0 101842 010626 BIKF ELLX B757 ICE614 1 1 0 2977 372

```



### 5.2.40 pnl - Penalisation file

penalisation	
extension:	pnl
origin:	produced manually
separator:	one of more spaces (ASCII character 32)
sort:	no sort
comment:	each line contains the penalisation for a network segment. It is possible to have a comment line starting with "/"

#	Field	Type	Size	Comment
1	Point 1	char	26	the starting point of the segment to penalize
2	Point 2	char	26	the ending point of the segment to penalize
3	penalisation	num		penalisation (in NM)

#### Example :

```
LONTA  BEDAK 1000
LONTA  GONAR 1000
MODRA  KOGAT 1000
LINDU  OLOTA 1000
VAL     KOGAT 1000
```

### 5.2.41 pre - Presentation file

Presentation file is a text file that contains names and possibly path of several TDV files. Once a PRE file is loaded, the user can navigate from one TDV to the next either manually or automatically.

presentation	
extension:	pre
origin:	produced manually
separator:	one of more spaces
sort:	no sort
comment:	each line contains name and possibly path to a TDV file.

#	Field	Type	Size	Comment
1	name and possibly path to a TDV file	char		

#### Example :

```
EB_TMA.tdv
ED_TMA.tdv
LE_TMA.tdv
LF_TMA.tdv
LS_TMA.tdv
```

### 5.2.42 qt5 - Quick T5 file

Quick T5				
extension:	QT5			
origin:	produced by SAAM 3D Density			
separator:	blank			
sort:	no sort			
comment:	each line contains best flight trajectories 3D cell penetration			
#	Field	Type	Size	Comment
1	ID	num		flight ID found in field 17 of SO6
2	sector/cell name	char		numeric code representing the name of the cell

**Example (flight 1000 is crossing cells 3583, 3676 and 3770)**

```
1000 3583
1000 3676
1000 3770
```

### 5.2.43 rlex - Route Length Extension Exception

Route Length Extension Exception				
extension:	rlex			
origin:	produced manually			
separator:	blank (space or tab)			
sort:	no sort			
comment:				
#	Field	Type	Size	Comment
1	ADEP	char	4	Departure airport of the route
2	ADES	char	4	Arrival airport of the route
3	allowed extension	num		Allowed extension for this airport pair (in NM)

**Example :**

```
DGAA KJFK 500
DGAA LPAZ 180
DNAA LTAI 180
DNCA LPPT 170
DNKN GMMN 450
DNMM KJFK 200
DNMM LPFR 250
DNMM LTBA 190
DNMM TJSJ 1200
EBAW EBBR 75
EBAW EBZW 100
```

### 5.2.44 seg - Segment file

seg	
extension:	seg
origin:	used as an input for 3D density
separator:	blank
sort:	NO
comment:	each line contains a 3D segment with its associated load

#	Field	Type	Size	Comment
1	segment ID	num		uniq for a given 3D path
2	lat1	char		in minute decimal
3	lon1	float		in minute decimal
4	lat2	float		in minute decimal
5	lon2	float		in minute decimal
6	lev1	float		unit is FL
7	lev2	float		unit is FL
8	load	float		could be a floating number

#### Example:

```
97930540 2862.1 2123.3 2863.4 2116.3 56 38 229.647
97930540 2863.4 2116.3 2872.0333 2118.9833 38 0 167.57
97930541 2508.0167 734.3333 2508.2333 734.25 0 30 10.1385
97930541 2508.2333 734.25 2489.6333 729.6833 30 154 5219.27
```

### 5.2.45 sel - Sector Entry List

Sector Entry List	
extension	sel
origin	Export / SAAM to IPAS
separator	semicolon
sort	by callsign, but from Excel any other sort are possible
comment	Describes the sequence of crossed sectors (with entry level and entry time) flight per flight

#	Field	Type	Size	Comment
1	callsign	char		
2	origin	char		ICAO airport code
3	destination	char		ICAO airport code
4	aircraft type	char		ICAO aircraft type
3 x Fields below are repeated a certain number of times (on the same line) and sorted by increasing entry time				
5	entry time	char		formatted: HH:MM:SS
6	entry level	char		formatted: FLxxx
7	sector name	char		

#### Example (2 sectors are crossed):

```
BAW1234;EGLL;LFMN;B738;12:34:10;FL247;EGABC;13:05:59;FL340;LFFVU
```

## 5.2.46 sls - Sector List file

Sector List	
extension:	sls
origin:	Airspace Editor
separator:	blank
sort:	grouped by sector name
comment:	input of SAAM Traffic intersection / Airspace Editor ... The link between this and newmaxo ascii region file ARE is made by volume name.

NORMAL LINE				
#	Field	Type	Size	Comment
1	sector name	char		max 19 characters
2	volume sign	char	1	always "+" indicates that the volume has to be added to the sector. Note that subtraction is deprecated and should not be used any longer.
3	volume name	char		refer to name of volume in the <a href="#">ARE</a> file (max 24 characters).
4	volume bottom level (*)	num	1 - 3	in FL, overwrite level present in <a href="#">ARE</a> file for that volume (leading 0s are not mandatory).
5	volume top level (*)	num	1 - 3	in FL, overwrite level present in <a href="#">ARE</a> file for that volume (leading 0s are not mandatory).

Note: previous sls format accepted meta line, this feature is not supported anymore and can not be used, instead use ".spc" format.

### Example:

```
CASTOR_UP + AX 000 295
CASTOR_UP + GU 245 265
TMA + SLP 000 095
TMA + SLX 000 125
ABRON + TOP 45 660
```

## 5.2.47 slt - Graph filter file

Graph filter file				
extension	slt			
origin				
separator	no separator since field 1 is the complete line.			
sort	any order			
comment	this file contains a list of sectors (one name per line). It is used by <a href="#">SAAM chart</a> to hide the graph of the sectors that are not present in the filter file.			
#	Field	Type	Size	Comment
1	sector name	ascii string compose of letters (a-z A-Z), digits (0-9) and special characters (_-@)		Characters after the # are considered as comment
	partial sector name: begin with			
	comment	ascii characters		First character must be #

**Example:**

```
# A comment always start with #
EB # All sectors beginning with EB
LFFF
EDDD
```

**5.2.48 soa - Area Definition file**

Area Definition				
extension	soa			
origin	SAAM			
separator	blank (space or tabulation)			
sort	an area has to be defined prior to any usage in another area definition (see Roissy in example below)			
comment	used to simplify segment management rule writing for SAAM assignment. an area is a Set of Airports. Area Definitions appear in RAD appendix 2.			

#	Field	Type	Size	Comment
1	Area name	char	>= 5	name used in segment management rules. Name must be at least 5 characters long.
2..N	Airport or sub-Area name	char	>= 4	list of airport or sub-area names, blank (space or tabulation) separated. When a sub-area is used, the associated list of airport names is used recursively. Airport name may contain "." character to replace any [A-Z] character, like in "ED.."

**Example:**

```
Ajaccio LFKF LFKG LFKJ LFKO
Amsterdam EHAM EHHV EHKD EHLE EHRD EHTX
Roissy LFPB LFPC LFPG LFPT
Paris Roissy LFPN LFPO LFPV
Italy LI..
```

**5.2.49 so6 - Traffic file**

SegOut6	
extension:	so6
origin:	SAAM profile processing or PRISME Database or DDR
separator:	blank
sort:	by flight segment sequence from origin to destination
comment:	Describes flight 4D trajectories, 4D segment per 4D segment. This trajectory can comes from different sources: - SAAM (trajectories could be shortest, cheapest, optimal ...). SAAM calculates vertical profile possibly using operational constraint (from .flc2 file) and mix of BADA & CFMU aircraft performance. - CFMU (trajectories could be: M1=Model 1=last filled flight plan, M3=Model 3=flight plan

	<p>enhanced with Radar Data, ...). It should be noted that CFMU processes the vertical profile from flight plan using possible operational constraint and their own aircraft performance.</p> <ul style="list-style-type: none"> <li>- pure Radar Data (called CPR data)</li> <li>- other sources.</li> </ul> <p>This file could also contains flight options (flight ID &gt;= 1000000000), in such case it should be accompanied with a ".lox" file.</p> <p>This files is used as an input in many other SAAM processes.</p>
--	---

#	Field	Type	Size	Comment
1	segment identifier	char		first point name "_"last point name (see noite)
2	origin of flight	char	4	ICAO code
3	destination of flight	char	4	ICAO code
4	aircraft type	char	4	
5	time begin segment	num	6	HHMMSS padded with o's
6	time end segment	num	6	HHMMSS padded with o's
7	FL begin segment	num	1 to 3	
8	FL end segment	num	1 to 3	
9	status	char	1	0=climb, 1=descent, 2=cruise
10	callsign	char		
11	date begin segment	num	6	YYMMDD padded with o's
12	date end segment	num	6	YYMMDD padded with o's
13	lat begin segment	float		in minute decimale
14	lon begin segment	float		in minute decimale
15	lat end segment	float		in minute decimale
16	lon end segment	float		in minute decimale
17	flight identifier	num		same as the one provided in expand file (must be uniq). In case of flight option it is >=1000000000)
18	sequence	num		start at 1 for every new flight, incremented at each lines. IMPORTANT !
19	segment length	float		in nautical miles
20	segment parity/color	num		0=NO (grey, R=102, G=102, B=102), 1=ODD (green, R=60, G=255, B=60), 2=EVEN (blue, R=100, G=100, B=255), 3=ODD_LOW (dark green, R=0, G=200, B=0), 4=EVEN_LOW (light blue, R=160, G=160, B=255), 5=ODD_HIGH (light green, R=160, G=255, B=160), 6=EVEN_HIGH (dark blue, R=0, G=0, B=200), 7=General Purpose Red Color (R=255, G=0, B=0), 8=General Purpose Orange Color (R=255, G=128, B=0), 9=General Purpose Yellow Color (R=255, G=255, B=0)

Note: point name are 3 or 5 characters for route points and 4 characters for airports. Route point starting with a \$ or % are special

SAAM point. CFMU route point having coordinates but no name should be identified with ! followed by 4 digit like: !0034

**Example (2 flights):**

```

DAAG_ALR DAAG LEBL B727 150000 150045 0 21 0 DAH2012 970905 970905 2202 193 2200 190 1
ALR_$aaaa DAAG LEBL B727 150045 151725 21 330 0 DAH2012 970905 970905 2200 190 2298.33
$aaaa_MHN DAAG LEBL B727 151725 153013 330 320 2 DAH2012 970905 970905 2298.33 221.72
MHN_$aaab DAAG LEBL B727 153013 153632 320 320 2 DAH2012 970905 970905 2391 253 2422.7
$aaab_TASOS DAAG LEBL B727 153632 154356 320 145 2 DAH2012 970905 970905 2422.74 207.2
TASOS_QUV DAAG LEBL B727 154356 155155 145 4 2 DAH2012 970905 970905 2456 158 2477 125
QUV_LEBL DAAG LEBL B727 155155 155213 4 0 2 DAH2012 970905 970905 2477 125 2477 124 11
EBBR_BRU EBBR EKBI B73V 173000 173100 0 18 0 DAN258 970905 970905 3054 269 3054 262 46
BRU_NIK EBBR EKBI B73V 173100 173516 18 93 0 DAN258 970905 970905 3054 262 3070 251 46
NIK_%EH21 EBBR EKBI B73V 173516 173842 93 150 0 DAN258 970905 970905 3070 251 3086 241
%EH21_SPY EBBR EKBI B73V 173842 174958 150 285 0 DAN258 970905 970905 3086 241 3152 29
SPY_GRONY EBBR EKBI B73V 174958 175618 285 339 0 DAN258 970905 970905 3152 291 3183 34
GRONY_$aage EBBR EKBI B73V 175618 180012 339 370 0 DAN258 970905 970905 3183 346 3197.
$aage_BEDUM EBBR EKBI B73V 180012 180059 370 370 2 DAN258 970905 970905 3197.06 386.39
BEDUM_NL1 EBBR EKBI B73V 180059 180320 370 370 2 DAN258 970905 970905 3200 395 3213 41
NL1_$aagf EBBR EKBI B73V 180320 180802 370 370 2 DAN258 970905 970905 3213 415 3246.62
$aagf_VES EBBR EKBI B73V 180802 182220 370 103 2 DAN258 970905 970905 3246.62 436.807
VES_EKBI EBBR EKBI B73V 182220 183034 103 0 2 DAN258 970905 970905 3336 498 3344 549 4

```

**5.2.50 sos - Sector Boolean Opening Scheme file**

Sector boolean Opening Scheme	
version:	1
extension:	.sos
origin:	SAAM, input & output from optimisation
separator:	tab
sort:	no sort
comment:	Describes sector per sector, possible opening time, independently from other sectors (overlap possible in a given sector family) Number of columns is variable (below n* is 25 for 24 periods of 1 hour, or 49 for 48 periods of 1/2h, or 97 for 96 periods of 1/4h)

Header				
#	Field	Type	Size	Comment
1	blank		o	nothing here
2	time	char		time expressed HHMM (zero padded)
...	time			
n*	time	bool		time expressed HHMM (zero padded)

Body (Matrix)				
#	Field	Type	Size	Comment
1	sector name	char		could be elementary sector or collapse sector, must be found in .cfg file
2	opening info	bool		1 means open for this period of time, 0 means closed for this period of time
....	opening info	bool		1 means open for this period of time, 0 means closed for this period of time
n*	opening info	bool		1 means open for this period of time, 0 means closed for this period of time

**Example (24 periods of 1 hour; 6 sectors; displayed via excel with coloured cells):**

C1_285_355	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
C1_285_660	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
C1_355_660	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
C1_C2_285_660	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
C1_C2_Lshape	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	0	0	0
C1_C2_U_285_355	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

### 5.2.51 spc - Gasel Airspace file

Airspace	
extension	spc
origin	Gasel, SAAM
separator	;
sort	no sort, this file could be self referenced
comment	Describe how elementary sectors can be collapsed, can also contains description other bigger airspace (like ACC ...)

#	Field	Type	Size	Comment
1	nb	num	1	number of airspace structures defined
				Header
#	Field	Type	Size	Comment
1	tag	char	1	must be "A"
2	name of airspace	char	~	ID, used in other files, like cfg
3	comment	char	~	free text descipbing the airspace, can be empty
4	type of airsapce	char	~	"CS" for collapse, "AUA", "AUAG", "RSA" ...
5	number of items	num	~	number of items composing the airspace (indicate the number of body lines)
Body, made with Header "number of items" lines				
#	Field	Type	Size	Comment
1	tag	char	1	must be "S"
2	name of airspace	char	~	ID, used in other files, like sls, or somewhere else in this file
3	type of airsapce	char	~	"ES" for elementary sectors, could be "AUA" or other

### Example: 5 airspaces

```

5
A;EKDKECTA;COPENHAGEN ACC EAST;AUA;8
S;EKDK5;ES
S;EKDK6;ES
S;EKDKA;ES
S;EKDKB;ES
S;EKDKC;ES
S;EKDKD;ES
S;EKDKE;ES
S;EKDKI;ES
A;EKDKLV;CPH SECTOR L+V;CS;2
S;EKDKL;ES
S;EKDKV;ES
A;ENLECS;SPAIN SEVILLA;AUAG;5
S;LEAMTMA;AUA
S;LEBZCTR;AUA
S;LECSCTA;AUA

```



```

S;LEGRMTMA;AUA
S;LEMGMTMA;AUA
A;ENKBTMA;KVERNBERGET TOWER;AUA;1
S;ENKBTA;ES
A;LBD228;_;RSA;1
S;LBD228;ES

```

### 5.2.52 ssl - SID and STAR list

Airspace	
extension	ssl
origin	IPAS
separator	blank
sort	no sort
comment	Optional management of suppression of SID/STAR points to/from airports within the area to be simulated is triggered via a user input text file having .ssl extension (if empty the SID/STAR management does not take place, but the conversion to IPAS will be valid)

#	Field	Type	Size	Comment
1	airport_name	char	4	number of airspace structures defined
2	D or A	char	1	A or D for Arrival or Departure
3	point_name	char	~	last point of the SID or first point of the STAR

#### Example:

```

LSGG D JOHAN

```

### 5.2.53 t5 - Airspace Traffic Intersection file

T5	
extension	T5
origin	produced by SAAM Airspace Traffic Intersection
separator	blank
sort	Important ! 1/ field 1 (flight ID), 2/ field 2 (sector name), 3/ field 3 (entry date/time) as numeric value
comment	each line contains entry and exit information for a flight for a sector (n lines for n entry / exit of the same flight in the same sector)

#	Field	Type	Size	Comment
1	ID	num		flight ID
2	sector name	char		
3	entry date/time	float		unit is second, starting 01/01/1970
4	exit date/time	float		unit is second, starting 01/01/1970
5	entry FL	float		unit is Flight Level decimal
6	exit FL	float		unit is Flight Level decimal
7	entry distance	float		unit is nm decimal, from beginning of segment
8	exit distance	float		unit is nm decimal, from beginning of segment
9	entry segment name	char		separator between point name is "_" "##" as a prefix means that the flights was already present in the sector before it was opened
10	exit segment name	char		separator between point name is "_", might be different from field #9 "##" as a prefix means that the flight was still present in

				the sector after it was closed
11	total distance spent in sector	float		for that flight ID for that sector, expressed in nm decimal
12	total time spent in sector	float		for that flight ID for that sector, expressed in second decimal

**Example:**

```

16883 LQSB 2197702.791411 2198377 245.000000 4.000000 23.503334 90.142200 $amqx_DBK $a
16964 LQSB 2182545.461068 2183300.537904 230.000000 175.296125 53.386000 12.871500 ZAG
17262 LQSB 2204806.486014 2205400.493804 245.000000 62.522849 12.452444 62.947700 $amE
17546 LQSB 2202699.898361 2202784.662148 245.000000 206.642500 14.200000 21.871500 MIR
18151 LQSB 2202533.576397 2204697.097729 240.000000 240.000000 21.281700 105.638700 VA
19454 LQSB 2213785.364431 2215008.500000 240.000000 245.000000 14.792400 48.395400 $ao

```

**5.2.54 tax - Turn Angle Exception file**

penalisation	
extension:	tax
origin:	produced manually
separator:	blank (space or tabulation)
sort:	no sort
comment:	the list Point1 Point2 Point3 written on each line defines the Point1_Point2 Point2_Point3 turn

#	Field	Type	Size	Comment
1	Point 1	char	26	defines with Point1 the incoming segment of the turn
2	Point 2	char	26	the turn point
3	Point 3	char	26	defines with Point2 the outcoming segment of the turn

**Example :**

```

ADOKI SKAVI KRW
ADOSA VIL02 VIL.A
ADUSO BENAK POBIX
ADUTO FERDI OKT.A
ADX POS MUROS
AGB.D RIDAR LELTA

```

**5.2.55 tcr - T5 Cleaning Rule file**

T5 Cleaning Rule	
extension	tcr
origin	SAAM
separator	TABULATION !!!
sort	important
comment	ascii file describing the rule to suppress flights from some airspace, works on T5 file

#	Field	Type	Size	Comment
1	rule ID	num		rule identifier
2	line ID	num		0 is header, 1 is free comment, 2 is flight query criteria (can be multi lines), 3 is list of airspace name
3	info	variable		blank separated, content depends on line ID (see notes)

**Notes:****Line ID=0 (header) fields are**

- version number (=0),
- toggle (=1 means rules is activated, =0 means rules disabled)
- date creation of the rule (DD/MM/YYYY)
- time creation of the rule (HH:MM:SS)

**Line ID=1 (free comment):**

- contains what user puts, can be used to search a rule

**Line ID=2 (flight query criteria):**

- contains the exact values of query criteria, used to identify the flights that must be deleted from airspace
- In case of a qif criteria several line ID=2 will be present

**Line ID=3 (list of airspace name):**

- contains the list (blank separated) of airspace from which the set of identified flights will be deleted

**Example (2 rules)**

```
1 0 0 1 08/01/2004 18:01:11
1 1 Depart Bruxelles
1 2 EBBR EG??
1 3 EBBUEST
2 0 0 1 09/01/2004 10:44:12
2 1 Depart NICE
2 2 LFMN_
2 2 BLONA
2 3 LFMMTOP LFMMSOUTH
```

**5.2.56 tdv - Three Dimensional Viewer file**

Three Dimensional Viewer	
extension	tdv
origin	file used to describe a 2D/3D/4D scene to visualized with SAAM
separator	blank
sort	no, but order is important
comment	ascii file describing shapes used as an input in SAAM for 3D terrain display

#	Command type	TDV code	Parameters		
1	Airspace	V	S/T(*)	L N T OB X CI X (top, border, inside feature *)	file_name.are(*) bot_level top_level COLOR(*)
	Airspace	Vx	version(=1) S/T(*) L N T OB X CI X (top, border, inside feature *) file_name.are(*) bot_level top_level COLOR(*) border_thickness border_COLOR(*)		
2	3D Line	L	S/T(*)	segment_name, line shape(*),lat1,lon1,alt1,lat2,lon2,alt2,size, COLOR(*)	

3	3D Point	P	S/T(*)	point_name, point shape(*), lat, lon,alt_base,height,size (*),COLOR (*)				
4	3D Designator	D	text_t o_ dis pla y	F/B/T/X(not implemented)	lat	lon	alt	COLOR(*)
5	3D Designator	Dx	version(=1) text@to@@display 2nd@line lat lon alt feature_version(=1) aspect(=F/B/T/X*) frame_thick(=0/1) text_alignment(=L/C/R) post(=0/1) post_shiftX post_shiftY post_alignment(L/C/R) font_logical_name(*) auto_color_text(0/1) COLOR_text(*) auto_color_frame(0/1) COLOR_frame(*) auto_color_backgrd(0/1) COLOR_background(*)					
6	font	FONT	version(=1) font@logical@name size(in point, max 200) weight (100=thin, 400=normal, 900=heavy) bool_italic(0 or 1) bool_underline(0 or 1) bool_strikeout(0 or 1) charset(ANSI or SYMBOL) system@font@name					
7	Camera rotation (unique object)	XORIENT	angle_rotation_movement(0 means stable) angle_rotation_value(in tens of degrees decimal, approximately 0 means top of screen is north oriented, 450 means top of screen is west oriented). Caution: this is a pure euclidian angle (not north oriented). A tilde value for any of the 2 parameters means current value is unchanged.					
8	Camera tilt (unique object)	YORIENT	tilt_angle_movement(0 means stable) tilt_angle_value(in tens of degrees decimal: 0=top view, -450 side view) A tilde value for any of the 2 parameters means current value is unchanged.					
9	Camera zoom (unique object)	ZOOM	zoom_movement (0 means stable) zoom_value (0 means eye touching the object, 1000 view airport, 10000 view approach, 40000 view small country, 100000 view large country, 280000 view europe) A tilde value for any of the 2 parameters means current value is unchanged.					
10	Camera E/W translation (unique object)	XMOVE	x_translation_movement (0 means stable) x_translation_value (west-east oriented, +value means goes to west, -value goes to east, independently from rotation angle) A tilde value for any of the 2 parameters means current value is unchanged.					
11	Camera N/S translation (unique object)	YMOVE	y_translation_movement (0 means stable) y_translation value (south-north oriented, +value means goes to south, -value means goes to north, independently from rotation angle) A tilde value for any of the 2 parameters means current value is unchanged.					
12	Camera vertical ref (unique object)	YVMOVE	ground_ref_movement (0 means stable) ground_ref_value (0 means vertical reference is the ground, -value goes above ground, +value goes below ground) A tilde value for any of the 2 parameters means current value is unchanged.					
13	Camera changes absorption (unique object)	AMORT	XORIENT YORIENT XMOVE YMOVE ZOOM YVMOVE (movement absorber, value with decimal, 99999 means stop movement, 0 means never stop movement, +value means stop gradually, -value means accelerate gradually if movement value != 0). These values change the movement parameter of the associated objects. A tilde value for any of the 4 parameters means current value is unchanged.					
14	Cube	C	S/T	name, lat1,lon1, lat2,lon2,lat3,lon3,lat4,lon4, bot_level, top_level, COLOR				
15	Flight Animation (old version)	SANIM	file_to_animate.so6 FROM_MEMORY FROM_SOCKET(*) aircraft_shape (*) aircraft_size(*) COLOR(*). Caution: does not handle new "Custom Cylinder Height" aircraft shape.					
16	Flight Animation (new version)	SANIMx	version(=1) file_to_animate.so6 \\.\pipe\\mynamedpipe FROM_MEMORY FROM_SOCKET(*) aircraft_shape(*) aircraft_size(*) COLOR(*) volume_name_to_be_moved(else NULL; if used it requires aircraft_shape value to be 0 and a single flight in the so6 file)					

			Caution: version 1 does not handle new "Custom Cylinder Height" aircraft shape.  version (=2) file_to_animate.so6\\.\pipe\\mynamedpipe FROM_MEMORY FROM_SOCKET(*) aircraft_shape(*) aircraft_size(*) COLOR(*) volume_name_to_be_moved(else NULL; if used it requires aircraft_shape value to be 0 and a single flight in the so6 file)			
17	Density Anim	DENS	ori_lat ori_lon ori_lev size_airspace(East,North,Height) size_cell(E/N/H) count_meth(0 or 1 or 2) para thresh(R,O,Y,G) Notes: - a single command should be present in a TDV, and should appear after SANIM TDV command (due to coef processing) - count_meth: 0=exact count with classical cell display, 1 count is made with a trailing average of para length, 2 exact count with smooth color display			
18	Clock Anim (unique object)	CLOCK	start(in s) end(in s) step(in s decimal) clock_wrap(=LOOP or NOLOOP) Note: see comment below for NOLOOP.			
19	Clock Anim (unique object)	CLOCKx	version (=1) start(*) end(*) [z]step(in s decimal) clock_warp(=LOOP or NOLOOP) display_status (default is NTS*) If step parameter is preceeded with "z" time speed animation is Zoom dependant (quicker with zoom out, slower with zoom in, same speed with europe scale "HOME" pressed) If NOLOOP is set the animation stops when clock reached the end, but this mode might imply : - if presentation mode is ON with timer set to 0, clock reaching end of animation will open next TDV file automatically, allowing synchronisation of events (with following time value or not) amongst different TDV files.			
20	Clock pause (unique object)	PAUSE	(no parameters, comment: linked to keyboard pause key, freeze the clock)			
21	Terrain	TERRAIN	file_name			
22	Vertical magnifier (unique object)	COEF	altitude_coef (default is 50, if used must appear before any other TDV command)			
23	Background white color (unique object)	WHITE	Note: if used with			
24	Background Color (unique object)	BACK_COLOUR	COLOR(*)			
25	Window frame screen position (unique object)	PREF_POSITIONS	x1	y1 x2 y2 (command valid in former Unix version, not yet implemented on Windows)		
26	Lambert project para (unique object)	LAMBPARA	unit (DEG / MIN)	latN latS lonE lonO		
27	Airport	RUNWAY	name	lat1 lon1 elev1 (in FL)	(not used)	elev2 (in FL) stop2 (not

		Y		stop1 (not used)	lat1 lon2	used) larg (in nm) COLOR
28	Line comment	//				
29	start block comment	/*				
30	end block comment	*/				
31	Route Network	NETWOR K	versio n (= 1)	display(=LOAD PARITY) ase_file network_COLOR(*)		
32			versio n (= 2)	LOAD/PARITY ase_file red_thresh ora_thresh green_thresh blue_thresh min_grey_thresh network_COLOR(*)		
33			versio n (= 3)	display(=DEFAULT PARITY LOAD TYPE MIXED) ase_file red_thresh ora_thresh green_thresh blue_thresh grey_thresh size_point size_segment size_arrow label_on_off label_size label_frame network_COLOR		
34			versio n (= 4)	display(=DEFAULT PARITY LOAD TYPE MIXED) ase_file red_thresh ora_thresh green_thresh blue_thresh grey_thresh size_point size_segment size_arrow label_on_off label_size label_frame label_load network_COLOR(*)		
35			versio n (= 5)	display(=DEFAULT PARITY LOAD TYPE MIXED) ase_file red_thresh ora_thresh green_thresh blue_thresh grey_thresh size_point size_segment size_arrow label_on_off label_size label_frame label_load nb_load_decimal network_COLOR(*)		
			versio n (= 6)	display(=DEFAULT PARITY LOAD TYPE MIXED) ase_file red_thresh ora_thresh green_thresh blue_thresh grey_thresh size_point size_segment size_arrow point_label_on_off load_label_on_off load_label_nb_deci network_COLOR(*) point_label_feature={version(=1) aspect (=F B T X) thick text_align(=L R C) post_on_off post_x post_y post_align(=L R C) logical_font_name text_auto_color_on_off text_COLOR(*) frame_auto_color_on_off frame_COLOR(*) background_autocolor_on_off background_COLOR(*)} load_label_feature={version(=1) aspect(=F B T X) thick text_align(=L R C) post_on_off post_x post_y post_align (=L R C) logical_font_name text_auto_color_on_off text_COLOR(*) frame_auto_color_on_off frame_COLOR(*) background_autocolor_on_off background_COLOR(*)}		
			versio n (= 7)	display(=DEFAULT PARITY LOAD TYPE MIXED CDR) ase_file red_thresh ora_thresh green_thresh blue_thresh grey_thresh size_point size_segment size_arrow point_label_on_off load_label_on_off load_label_nb_deci network_COLOR(*) point_label_feature={version(=1) aspect (=F B T X) thick text_align(=L R C) post_on_off post_x post_y post_align(=L R C) logical_font_name text_auto_color_on_off text_COLOR(*) frame_auto_color_on_off frame_COLOR(*) background_autocolor_on_off background_COLOR(*)} load_label_feature={version(=1) aspect(=F B T X) thick text_align(=L R C) post_on_off post_x post_y post_align (=L R C) logical_font_name text_auto_color_on_off text_COLOR(*) frame_auto_color_on_off frame_COLOR(*) background_autocolor_on_off background_COLOR(*)}		

			<p>label_point_min_converg_seg label_point_max_converg_seg  label_point_min_diverg_seg label_point_max_diverg_seg  label_load_min_value label_load_max_value</p>
36	Diagram	DIAGRAM	<p>title Y_axe_70%_written_value(max 3 digits) position(0=left side, 1=right shifted, 2=more right shifted, 3=right maxi) COLOR</p>
37	Diagram	CURVE	<p>title data_file_name(file extension ".tim" have a info line header)  attached_diagram(0 to 3) text_position(1 to 6) data_file_column(*)  factor(*) curve_display_flag(Y or N used with DIAGRAM object)  attached_object(*see explanation below) COLOR(*)</p> <p>Note: if the second column of ".tim" file (linked to this CURVE object) is called in the header (first line) "NB_CLOCK_TURN", the animation will be effective only if the number of clock turns (calculated internally by SAAM) matches the value found in the second column. This particular column does not affect data_file_column positionning.</p>
38	title on screen	TITLE	<p>version=1 aspect(S/T) text_to_display(blank=@) size_of_the_font  screen_coord_x screen_coord_y COLOR</p> <p>version=2 aspect(S/T) text_to_display(blank=@, one '#'=numbering part*coef coming from CURVE for animation, or one '^'=current clock time does not more require CURVE) size_of_the_font screen_coord_x  screen_coord_y shadow_x shadow_y COLOR(*)</p> <p>version=3 aspect(S/T) test_to_display(blank=@, one '#'=numbering part*coef coming from CURVE for animation, or one '^'=current clock time does not more require CURVE) destination(3D_SCENE or SCREEN) font_system_name(like "Arial") extrusion(depth of the character, 0 for SCREEN) deviation(not used should be 0) size(0 is extremely small, visible size should be around 1000.0, same value as version previous version for SCREEN) latitude(located in the middle of the text for 3D_SCENE or X coordinates on the left of the text for SCREEN) longitude(located in the middle of the text for 3D_SCENE or Y coordinates for SCREEN) altitude (in flight level, located in the middle of the text for 3D_SCENE, 0 for SCREEN) rotation_X(expressed in degrees, can be over 360.0, title turn around horizontal axe) rotation_Y(expressed in degrees, can be over 360.0, title turn around vertical axe) rotation_Z(expressed in degrees, can be over 360.0, title tilt) shadow_x(only for SCREEN, 0 for 3D_SCENE) shadow_y(only for SCREEN, 0 for 3D_SCENE) COLOR(*)</p>
39	picture	IMAGE	<p>version=1 aspect(S/T; T is not used) file_name.bmp size_factor(in decimal; 0=very very small, 1.0=original file size) lat lon alt (3D location is centered in the lower part of the picture)</p> <p>version=2 aspect(S/T; T is not used) 3D_SCENE(default) SCREEN  file_name.bmp size_factor(in decimal; 0=very very small, 1.0=original file size) lat lon alt</p> <p>Comment: if 3D_SCENE then bitmap is displayed in the scene (bitmap origin is centered in the lower part), else it is display on the top of the SCREEN and lat &amp; lon respectively are x and y in relative pixels (bitmap origin is lower left corner) and alt is not used.</p> <p>version=3 aspect(S/T; T is not used) 3D_SCENE(default) SCREEN  file_name.bmp size_factor(in decimal; 0=very very small, 1.0=original file size) origin(2 letters:x-axe={Left, Centre, Right}, y-axe={Top, Center, Bottom}, example: CB) lat lon alt</p> <p>Comment: if 3D_SCENE then bitmap is displayed in the scene (default bitmap origin is centered in the lower part), else it is display on the top of the SCREEN and lat &amp; lon respectively are x and y in relative pixels (default bitmap origin is lower left corner) and alt is not used.</p>

40	image is 2D (unique object)	2D	(by default TDV image is 3D, this command displays it in 2D at first)
41	comment	# or //	allows to comment and/or to disable TDV command
42	comment	/* then */	Allows to comment out TDV lines between /* and */. Must be alone on the line.
43	camera angle (unique object)	CAM_ANGLE	version(=1) angle(expressed in degrees decimal ranges from 1.0=no perspective to 179.0=very high distortion, default is 30.0)
44	2 views on screen (unique object)	SCREENS HARE	version(=1) TOP_SCREEN or BOTTOM_SCREEN or LEFT_SCREEN or RIGHT_SCREEN(used with presentation "pre" file, to display 2 TDV files in same time on the same screen)
45	Hidden surface display removal (unique object)	CULLFACE	No parameter. By default SAAM displays hidden surface, this command stops displaying them. In this case, and if your surface is anti-clockwise, it will be not displayed correctly. Note: special object (like transparent Title object in 3D scene) are not affected by this command.
46	Enable Stereo mode (unique object)	STEREO	version (=1) Note 1: will display the current TDV file in stereo (default is no stereo) Note 2: stereo parameters can be changed with shortcut ALT+Z (see keys/effect indications on screen)
47	Layer	LAYERSET	version(=1) status(0 or 1) readwrite(RO or RW) path_of_directory_containing_dataset Note 1: the last parameter contains a sub-directory called baseline which have all data files for the baseline (files of dataset coming from DDR) and *.prj files (one per proposal). Note 2: several LAYERSET commands are possible except if the same project is open in mode read-write by several SAAM, to avoid concurrent access problem (see c:\SAAM_USER_PREF\layer_lock.txt file) Note 3: this command must be written before any other command which reference a dynamic '0000' file.
48	Object hierarchy	GROUP	version(=1) display(0=hide, 1=show) skname(all object that belongs to the group will have this skname appended) Start TDV objects grouping, all TDV object after this GROUP will be together.
49	Object hierarchy	END	version(=1) end of previous GROUP. All objects before this END belongs to the same group.

(\*) size for Point only is expressed by default in internal unit, except when "nm" is added at the end of the size in that case it is expressed in nautical mile

### Aspect field for Designator (Dx command) means:

- F=Frame,
- B=Background,
- T=Two(frame+background),
- X=(no frame and no background)

**font\_logical\_name** is the logical name given in the FONT command else it can be "DEFAULT" for a Fixedsys500c.ttf font of size 12 normal thickness (which is installed automatically by SAAM)

**COLOR** = "C Transparent Red Green Blue" (each TRGB integer value can vary from 0 to 255)

Note: for ANIM or SANIM TDV object a Transparent value equals to 0 (zero) will display in 2D animated aircraft above all other transparent objects, and above the route network.



**lat** and **lon** are coordinates in minute decimal

**height**, **altitude** or **level** are in Flight Level decimal, size or distance are nautical miles decimal

#### **aircraft\_shape:**

- 0=no aircraft displayed,
- 1=a pixel,
- 2=flat triangle,
- 3=3D triangle,
- 4=height limited cylinder,
- 5=max height cylinder,
- 6=noise contour radius fixed,
- 7=noise contour radius varying,
- 8=B737 3D Model,
- 9=change camera look-at point,
- 10=change camera location. For camera changes (9 or 10), as well as volume\_to\_be\_moved different from NULL, it is recommended to use single flight so6 files (i.e one single flight so6 file for camera look at and one - possibly different - single flight so6 file for camera location)
- 12=Solar Impulse model. Note: codes running from 12 to 30 are reserved for generic manually created model. This model has always the label indicating its current FL displayed (with text "FL" in front the value).
- 1AAABBB. For custom cylinder height. The 3 digits (padded with 0's) AAA corresponds to the value expressed in FL above current aircraft FL, BBB corresponds to the value expressed in FL below current aircraft. AAA and BBB can be set to '000' but can not be negative. Example: 1050020 indicates a cylinder having a height of FL50 above the aircraft current FL and FL20 below.

**aircraft\_size:** expressed in floating value. If followed by nm (ex: 50.3nm) then expressed in nm (not valid for pixel shape, look-at point and camera location) if preceded with "z" aircraft size is Zoom dependant (bigger with zoom out, smaller with zoom in, same with Europe scale "HOME" pressed). Note: aircraft size factor may vary from aircraft shape, to be adjusted manually.

**FROM\_SOCKET** If this source is provided a named pipe is created and is used for reading ARTAS info on line instead of so6 file. The name of the pipe is: "\\.\pipe\mynamedpipe", with mynamedpipe that can be replaced by any name. Note: the writing pipe must be created before the animation is launched (see for instance prog like "writeToPipe" or "UDP\_Pieter"). Several different pipes can be opened into several different SANIM objects, but always a single pipe to a single SANIM object. Note: the case of night animation is not covered and will issue an error message (see clock time)

**S/T** means Solid or Transparent (TDV code is "S" or "T"), for other feature see below:

**Top feature code:** L=normal top not lighted, N=normal top lighted, T=solid top not lighted, O=solid top lighted

**Border** feature code: letter B (border has its own color, by default it is black), X (no border lines) or C (border line has the color of Inside) can be added at the end of top feature code

**Inside feature code:** letter I (default=inside is showed) or X (inside not displayed) can be added

at the end of border feature code

### Line shape:

*No arrow:*

- 0=horizontal line,
- 1=horiz. triangle,
- 2=horiz. square,
- 3=horiz. cylinder low def,
- 4=horiz. Cyl. med. Def.,
- 5=horiz. Cyl. high def.

*An arrow in the middle:*

- 20=horizontal line,
- 21=horiz. triangle,
- 22=horiz. square,
- 23=horiz. cylinder low def,
- 24=horiz. Cyl. med. Def.,
- 25=horiz. Cyl. high def.

*An arrow at the end of the line:*

- 40=horizontal line,
- 41=horiz. triangle,
- 42=horiz. square,
- 43=horiz. cylinder low def,
- 44=horiz. Cyl. med. Def.,
- 45=horiz. Cyl. high def.

### Point shape:

- 0=vertical line,
- 1=vertical triangle,
- 2=vertical square,
- 3=vertical low definition cylinder (8vertices),
- 4=vertical medium def. Cylinder (16 vert.),
- 5=vertical high def. Cylinder (32 vert.)
- 6=vertical up cone low def,
- 7=vertical up cone medium def.,
- 8=vertical up cone high def.
- 9=Up Starfish 3 branches,
- 10=Up Starfish 4 branches,
- 11=Up Starfish 5 branches,
- 12=Up Starfish 6 branches,
- 13=Up Starfish 8 branches

**CLOCK:** start end time parameter. By default expressed in seconds. If value begins with H then expressed in hours minutes seconds (like H134559 for 13h45m59s)  
if time start is greater than time end parameter than SAAM switch in "night" mode, and will assume the animation to run from start to midnight and midnight to end.

**Display Status:** 3 letters indicating how the 3 fields of the status line on screen is visible:

- In the order we have: number of aircraft='N', current time of animation (HH:MM:SS #nb clock turn) = 'T', current step='S'. Default is NTS.
- a 'X' instead of the field code hides the corresponding field. Examples: XTX: show only the time of animation, NXX shows only the number of aircraft animated ...

The "file\_name.are" of the TDV code "V" (volume) can be written all in upper case which means the volumes of the file are "hard" locked (no possible edition, no filtering ...),  
if at least one character is lower case, the volumes of the files can be edited or "softly locked" (temporary

user lock, filtering always available).

**CURVE data\_file\_column** parameter format is n[H][Rf] with:

- n column number where data must be read, ranges from 1 (column just after the time column) to max 100
- optional tag H means that time column is read with either HH or HHMM or HHMMSS or HHMMSS. decimal; else time column is in second decimal (default)
- optional tag R followed by parameter f means Random, with f is expressing a number of frame. It means that the value processed by SAAM is randomly created every f frame of animation, and ranges between 0 and data value read in data column n and then multiplied by factor (in between 2 frame calculations the latest value is kept). If the data value is 0 the random processing is disabled and the value is set to 0 (generally disabling the object)

### CURVE factor:

- Allows to expand or reduce vertical value read from tim file. Factor = 1 means max value found in selected column of the tim file are unchanged (so reaching top of diagram, if a diagram is used).
- If the diagram is used, the formula to calculate the factor value to reach => top of the diagram is:  
 $100 / (\text{max value found in the selected column in the tim file})$   
☐ => 70% line of the diagram is:  $70 / (\text{max value found in the selected column in the tim file})$   
☐ => equivalent to 70% written value is:  $70 / (70\% \text{ written value})$ . Warning: curve might overpass the diagram !!

### CURVE attached\_object\_name

attached\_object is either "NULL" (no object attached) or the name of an object (or several objects if the name are the same, depending on object) present in the TDV file (before or after CURVE command, see column 2 "object\_name" in the grid below)

When an object is attached, CURVE will change its appearance depending on the object type and on the value read from data\_file\_column multiplied by factor parameter found in CURVE parameter.

Below is a list of objects that can be attached with their possible graphical effect

TDV object	object_name	effect (n = value read fom data_file_column * factor)	comment
L (line)	segment_name	0=disable, n=enable + thickness	max 800 different L objects having the same segment_name can be linked into a single CURVE command
P (point)	point_name	0=disable, n=enable + thickness + size + height	max 800 different P objects having the same point_name can be linked into a single CURVE command
TITLE	text_to_display	a lot of possibilities (see <a href="#">TIM file</a> )	n value can be displayed in the title if '#' tag if found in "text_to_display" (idem for '^' tag for time display)
NETWORK	file_name.ase	0=disable, n=enable	
IMAGE	file_name.bmp	0=disable, n=enable + size of image	n=100 gives size given in IMAGE TDV command (if, in the IMAGE TDV command the size_factor is 1.0 then, with n=100, the size is the original)
D or Dx (designator)	text_to_display	0=disable, n=enable	
V or Vx (volume)	file_name.ave (means all sectors found in that file) <b>or</b> file_name.ave: sector_name (means	0=disable, n=enable + color change	linear interpolation between TRGB color components found in V or Vx object color and found in CURVE color. n=100 means full V or Vx color is displayed n=0 means full CURVE color is displayed

	only that sector found in that file)		Note: if RGB components are the same but only T (=transparency) is changed from 0 to 255, the object will appear/disappear on the screen.
SANIM (traffic animation)	file_name.so6	o=disable, n=enable (threshold at 0.5)	Allows to toggle traffic animation on/off
CURVE	title_of_curve	depends on object to which CURVE is linked (see above and below this row)	Several CURVE objects (number depends on memory size) having the same title_of_curve name can be linked into a single CURVE command. The value found in the 2 TIM files will be multiplied, and the result will be applied to animated objects.
Y MOVE (camera)	Y MOVE	n is used to drive the camera (move your scene as you wish, press F12 to add current camera values in the TDV, open the TDV and copy Y MOVE value)	goes in south/north direction (movement absorber is included)
X MOVE (camera)	X MOVE	n is used to drive the camera (move your scene as you wish, press F12 to add current camera values in the TDV, open the TDV and copy X MOVE value)	goes in west/east direction (movement absorber is included)
Y ORIENT (camera)	Y ORIENT	n is used to drive the camera (tilt your scene as you wish, press F12 to add current camera values in the TDV, open the TDV and copy Y ORIENT value)	change tilt angle (movement absorber is included)
X ORIENT (camera)	X ORIENT	n is used to drive the camera (rotate your scene as you wish, press F12 to add current camera values in the TDV, open the TDV and copy X ORIENT value)	change azimuth angle (movement absorber is included)
ZOOM (camera)	ZOOM	n is used to drive the camera (zoom as you wish, press F12 to add current camera values in the TDV, open the TDV and copy ZOOM value)	change zoom (movement absorber is included)

## 5.2.57 ter - Terrain Data file

Terrain data	
extension	ter
origin	SAAM terrain extraction module (terrain.exe)
separator	blank
sort	no, but order is important
comment	ascii file describing shapes used as an input in SAAM for 3D terrain display

#	Field	Type	Size	Comment
1	code	char		B, 'BT', 'N', 'C', 'V' or 'E'
2	parameters	list ...		depends on the code, can be empty

Code explanation with associated parameters:				
Cod e	Meaning	Parameters		Example
B	Begin triangle strip			B
BT	Begin independent triangle			BT
N	Normal Vector	X,Y,Z normalized components		N 0.494063 0.420676 -0.760877
C	Color	Red, green, blue component expressed between 0 and 255		C 180 255 180
V	Vertex	lat,lon,alt expressed in minute decimal/ meters		V 3355.2 -375 4.85564
E	End		E	

### Example:

```

B
N 0.174971 0 -0.984574
C 180 255 180
V 3349.8 -394.8 0
V 3355.2 -394.8 0
V 3349.8 -390 0.853018
N 0 -0.156032 -0.987752
V 3355.2 -390 0
N 0.107397 -0.155129 -0.982039
V 3349.8 -385.2 1.37795
N 0.046748 -0.207769 -0.97706
V 3355.2 -385.2 0.229659
N -0.106363 -0.206816 -0.972581
V 3349.8 -379.8 0.787402
N 0 -0.102739 -0.994708
V 3355.2 -379.8 0.229659
N 0.218922 -0.100247 -0.970579
V 3349.8 -375 1.87008
N 0.644724 0.369865 -0.668977
V 3355.2 -375 4.85564
N 0.494063 0.420676 -0.760877
E
B

```

## 5.2.58 tim - Timeline file

Timeline file	
extension	tim
origin	SAAM time file for animation
separator	tab → (space is also OK but then you <u>cannot</u> have a space in the column header row)
sort	rows must be sorted ascending by time (column 1) for one day or a part of a day
colons	First line = header which contains free comment or TOKEN (see below) The number of columns is variable: first column is for the time, and then 1 column for each type of animation. In addition there can be a special column to indicate the number of clock turns. If present, this column must be column 2.

The first row is a header row. It contains the column label. The label is not used by SAAM application (but necessary for the designer for commenting the content of the column) except for TITLE object, and number of clock turn that are all enabled via tokens stored in the header row. It is very important to do not put blank or tab within a header label, better use "\_" or "-" to separate words. Each TIM file is used in one or several CURVE object(s), [see CURVE format](#).

### List of recognised tokens in the header row of TIM file:

Number of clock turns:

- the token is NB\_CLOCK\_TURN, it must located imperatively in column 2, just after the time column. Useful for long animation, to start/end animations at Nth clock turn. Note: position of column referred in CURVE object are not changed by the presence or absence of NB\_CLOCK\_TURN column.

TITLE object (see [TITLE parameters](#)):

- DISPLAY to toggle on/off the display of the object (value is 0 or 1);
- LAT, LON, ALT to change respectively latitude, longitude and altitude (values are expressed in minutes decimals and in Flight Level, reference point is in the center of the Title);
- ROTX, ROTY, ROTZ will change angle of rotation for the different axes (values are expressed in degrees decimal, can be over 360.0 and negatives, note: order of rotation is important). ROTX rotates around horizontal x-axe ---, ROTY rotates around vertical y-axe | and ROTZ rotates around center z-axe o
- SIZE to change the overall size of the object (size 1.0 is extremely very small, visible size is about 1000.0);
- COLOUR to change the colour of the object. A value in the column should be between 0 and 100 to provide linear colour interpolation between the colour given to the Title object itself and the colour given to the CURVE object handling that animation, allowing to change one or several colour components: transparency, red, green and/or blue.

Notes:

- the values present in the column in the TIM file to change object properties are always linearly interpolated. If animation designer wants a sudden change than the time between the two states should be reduced to a minimum (0 or 1 second for example).
- If several animated events occurs at different times but are overlapping in the same TIM file, a convenient way is to use the special character "~" instead of manually calculating interpolated numeric value: it will be replaced by a linearly interpolated value processed internally (see example below)

### Example (time is expressed in seconds):

time	CIV_SIZE	MIL_SIZE
0	0	3000
1000	0	3000
2000	~	3000
2500	4000	~
4000	4000	0

In the example above a title object will grow linearly from size 0 at 13h to size 4000 at 20h. In the same time, a different title object will start to be reduced linearly from size 3000 at 14h to size 0 at 21h.

Data row (row 2 to ...)				
#	Field	Type	Size	Comment
1	Time	int/ float		time in seconds (max is 86400) or HH or HHMM or HHMMSS or HHMMSS.decimal (see <a href="#">CURVEparameter</a> to setup time format)
2	NB_CLOCK_TURN	int		this field is optional
2 or 3	value n for anim#1	int		
3 or 4	value n for anim#2	int		
....				
i or i-1	value n for anim#i			

### Example 1 (time is expressed in HHMMSS)

```
time   pinko_area my_designator
000000 0          0
133000 0          0
140000 100        0
143000 100        0
150000 100        1
235959 100        1
```

### Example 2: with a NB\_CLOCK\_TURN column token, 2 camera movements (free text), and one TITLE object token (SIZE)

```
time      NB_CLOCK_TURN      Cam_ZOOM      Cam_NS_move      SIZE
020000    2                  1246149    -27759           0
030000    2                  1246149    -27759           30000
060000    2                  300000     2258            10000
140000    2                  300000     2258            30000
```

For additional information on the timeline file, please refer to the [Creating a SAAM animation](#) topic.

## 5.2.59 tol - Bunching Capacity (Tolerance) file

Bunching capacity (tolerance)	
extension	tol
origin	
separator	blank
sort	no sort
comment	each line contains bunching capacity figures for a sector

#	Field	Type	Size	Comment
1	sector name	char	26	the same one as the one found in field 1 of sls
2	bunching capacity figure	int		special figure 999 means infinite capacity

Note 1: blank lines are allowed to better separate sectors from each others.

**Example:**

```
LFEPPLOW 3
EDDYLN0 5
```

**5.2.60 traj - Trajectory file**

Trajectory	
extension	traj
origin	produced by SAAM sector load (if traj check box is on), processed from t5 file and other flight information
separator	blank (except for field #1 which is composed with multiple information)
sort	On aircraft ID, then on route point order from origin to destination
comment	each line contains complete flight plan trajectories with sector entry and exit information

#	Field	Type	Size	Comment
1	aircraft ID	num		complete flight ID (format is: flightID_aircrafttype_callsign: origin_destination)
2	date/time	float		unit is s decimal, start 01 January 1970
3	flag	num	2	"-1=first flight plan point (departure) 0= sector exit point 1= normal flight plan point 2= sector entry point 3=last flight plan point (arrival)"
4	point or entry/exit sector name	char		if entry = in-sectorname, if exit = out-sectorname else route point name
5	latitude	float		unit is minute decimal
6	longitude	float		unit is minute decimal
7	level	float		unit is FL decimal

**Example (1 flight):**

```
17829_BA46_DLH5291:EPWA_EDDM 936974400 1 EPWA 3129 1258 0.00
17829_BA46_DLH5291:EPWA_EDDM 936974499 1 LIN2 3123 1263 53.00
17829_BA46_DLH5291:EPWA_EDDM 936974892 1 KRN2 3114 1213 189.00
17829_BA46_DLH5291:EPWA_EDDM 936975125 1 LDZ 3108 1179 255.00
17829_BA46_DLH5291:EPWA_EDDM 936975354 1 $aedG 3101.75 1143.25 300.00
17829_BA46_DLH5291:EPWA_EDDM 936976778.226916 2 in-CEATS_285_UNL 3057.751294 919.27069
17829_BA46_DLH5291:EPWA_EDDM 936976890 1 OKX 3054 902 300.00
17829_BA46_DLH5291:EPWA_EDDM 936977565 1 RAK 3006 822 310.00
17829_BA46_DLH5291:EPWA_EDDM 936977876 1 %LK2 2980 792 310.00
17829_BA46_DLH5291:EPWA_EDDM 936977996.737125 0 out-CEATS_285_UNL 2969.950521 780.5761
17829_BA46_DLH5291:EPWA_EDDM 936978152 1 KALOD 2957 766 297.00
17829_BA46_DLH5291:EPWA_EDDM 936978365 1 %ED43 2938 752 230.00
17829_BA46_DLH5291:EPWA_EDDM 936978661 1 MBG 2914 735 140.00
17829_BA46_DLH5291:EPWA_EDDM 936979132 1 EDDM 2887 701 0.00
```



# Index

## - \* -

\*.ase 146  
 \*.exp 145  
 \*.fld 151  
 \*.lox 147  
 \*.so6 17  
 \*.t5 46  
 \*.tdv 157

## - ' -

'OneSky Online' 16  
 'OneSky Teams' 16  
 'SAAM application' 16

## - 2 -

2D Assignment Rule 241  
 2D Options 39

## - 3 -

3D Constraints 241  
 3D Densities 78  
 3D Envelop 165  
 3D Options 39

## - 4 -

4D Options 39

## - A -

About SAAM 187  
 Add Airspace 133  
 Add Circle 141  
 Add Menu 132  
 Add Network 134  
 Add Point 216  
 Add Runway 137  
 Add Terrain 136  
 Add Traffic Animation 139  
 Adding a Title 239  
 Adding or Editing a Point 239

Adjust 20km 108  
 Airblock 196  
 Airblock file 17  
 Aircraft Category 241  
 Aircraft Label 190  
 Airport Arrival/Departure Time 101  
 Airports or Zones 241  
 Airspace 3D Envelop 165  
 Airspace data 17  
 Airspace Design 187  
 Airspace Editor 195  
 Airspace file 17  
 Airspace Flight Time 126  
 Airspace Load 90  
 Airspace PRU Complexity 124  
 Airspace to TDV 157  
 Airspace Value Merge 131  
 AIRSPACE VOLUMES 195  
 Airspace/Traffic Intersection 46  
 Analysis 10  
 animation clock features 190  
 are 360  
 ARE / SLS files 17  
 Are file 17  
 Areas to bypass 39  
 Arrival and Departure Constraints 241  
 Arrival and Departure Points 241  
 Arrival and Departure Rules 241  
 Assignment 28  
 Assignment Parameters 241  
 ATC TASKS 116  
 Axis properties 230

## - B -

Background Features 216  
 Basic 187  
 blank.tdv 16  
 Blue/White sea 190

## - C -

C: drive 16  
 CAPAN-Like 114  
 CDRs to bypass 39  
 cell 78  
 Centers 285  
 Centre of the Picture 21  
 Change Colour 196  
 Change Division Flight Level 196  
 Change name of airblock/sector 196

Change Text Font in Labels 216  
 Choosing PIN or ZIN 241  
 Clean T5 46, 70  
 Climb/descent constraints 39  
 clock 190  
 Collapsed sector 283  
 Combine Airspace Curves 97  
 Combine SO6, ARE or LOP 162  
 Command Line 360  
 Command line file 58  
 Command line info 187  
 Compare Airspace 107  
 Compare Network 104  
 Compare Traffic Routings 99  
 Complexity 131  
 Configuration Creation 290  
 CONFIGURATION EDITOR 274  
 Configuration file 17  
 Configuration Opening Scheme 17  
 Conflict 81, 160  
 Conflict file 81  
 Conflict search Tasks 116  
 Conflict Traffic Cut to Animation 160  
 Console 186  
 constraint file 42  
 Constraint\_not\_respected.txt 42  
 Continuous Picture Movement 21  
 Coordination Tasks 116  
 Copy 27  
 Copy and Paste airblock/sector 196  
 Courses 15  
 CRCO 108  
 Create Airblock/Sector 196  
 Create Segment 216  
 Criteria Browser 75  
 Criteria Options 67  
 Cut 27

## - D -

Data file 152  
 Data from Intranet (internal access) 181  
 DecimalMinutes 17  
 Default Scenario 24  
 Delay 122  
 Delete a Point 196  
 Delete Airblock/Sector 196  
 Delete Point 216  
 Delete Segment 216  
 Delta RFL 39  
 Delta time 39

Departure Constraints 241  
 Departure Points 241  
 Departure Rules 241  
 Desk 15  
 Direct Rte PIN 149  
 Disclaimer 14  
 Display Labels 196  
 Displaying the Network 216  
 DOS command prompt 360  
 Double Cut 196  
 down-load 16

## - E -

Edit mode 196, 216  
 Edit TDV file 190  
 Editing the Network 216  
 Editor 190, 195  
 Elementary sector 280  
 En-Route Constraints 241  
 En-Route Rules 241  
 Entry Rate 93  
 environment data 17  
 EUROCONTROL 10  
 executable 58  
 Export 10, 172  
 external access 181  
 Extract Traffic within Time Period 166  
 extranet 181

## - F -

File Menu 24  
 File Types 360  
 Filter 231  
 FL constraints option 39  
 FL Slice 196  
 Flight Data Management 116  
 Flight ID increase 162  
 Flow Demand to S06 151  
 Forbidden Segment 241  
 Forecast 51  
 Forecast file 51  
 Frame Features 216  
 Free Route 49  
 full screen 21  
 future traffic 51

## - G -

Gasel 46

Gasel format 17  
 Generic Launch 58  
 Get Basic Airspace Data 181  
 Graph 226  
 Graphical Airspace Editor 195  
 Graphical Slice Cut 196  
 Group 196  
 Group airblocks into sectors 196

## - H -

Hardware Requirements 13  
 Heavy Flight 42  
 Help 15, 187  
 Hide Airblock/Sector 196  
 HTML 187

## - I -

Import 10, 181  
 Information Line 72  
 Inserting a New Point 196  
 Inside Airspace Exact Cut 158  
 Inside Airspace First In Last Out Segment Cut 158  
 Installation 16  
 internal access 181  
 Intervention thresholds 116  
 Intranet 181  
 Introduction 10  
 IPAS 173

## - J -

Jump to the First Point 241

## - K -

Key file 152  
 Korn Shell 58

## - L -

Label Features 216  
 Lat/Long 17  
 Legend Set 187  
 Length/Azimuth 27  
 Level View Ref 186  
 Load of a segment 216  
 Lock 3D Airblock/Sector 196  
 Lowering S06 FL 156

## - M -

Manual 15  
 Map Features 24  
 Matching Files 152  
 Max number of options per flight 39  
 Max Route length extra extension 39  
 Maximum FL 89  
 Menu Bar 19, 24  
 Merge of airblocks 196  
 Merge Points 216  
 military area 52  
 Military Opening Time 52  
 Military Opening Time file 52  
 Minimum Distance Parameters 42  
 Model 1 17  
 Model 3 17  
 Modelling 10  
 Modifying the Assignment Parameters 241  
 Module 1 187  
 Module 2 187  
 Module 3 187  
 MOT Editor 270  
 MOT table 270  
 Move a Point 196  
 Move Point 216  
 Multi-Criteria Query 76

## - N -

Network 195, 241  
 Network data 17  
 Network info 190  
 new 24, 187  
 North Atlantic flights 241  
 North Compass 27  
 Number of RFL 39

## - O -

Open 24  
 Open Graph 24  
 Operating System 13  
 Optimisation 10  
 Optimum FL 42  
 Outside Airspace Exact Cut 158  
 Outside Airspace First In Last Out Segment Cut 158  
 Overload 96

## - P -

Page layout 234  
 Page Setup 24  
 Parameter list 58  
 parity of the segment 216  
 Paste 27  
 Paste airblock/sector 196  
 pdf 187  
 Perturbation file 81  
 Pick Info Menu 171  
 PIN 42, 148, 241  
 Point labels 196  
 Presentation 10  
 presentation files 190  
 Print 24  
 Print Preview 24  
 Printing graphs 238  
 Profile 42  
 Profile\_accurate.diag 42  
 PRU score file 124

## - Q -

QIF 76  
 Queries 66  
 Quit 24

## - R -

Radar Handovers 116  
 Radar Supervisions and Interventions 116  
 RAMS Airspace 176  
 RAMS Traffic 181  
 Random Distribution 42  
 Real Time 181  
 Recently used file list 24  
 Reference 15  
 Refresh 24, 190  
 Reserved Segment 241  
 RFL options 39  
 RFL Random Shift 42  
 RND SG 17  
 RND SG methodology 85  
 Rotate 21  
 Route Charge 108  
 Route Choice 52  
 Route Length 85  
 Route Option 39  
 Route Option Generator 39

RT Communications 116

## - S -

S06 to Network 146  
 S06 to Scenario 150  
 S06 to Text 144  
 SAAM 10  
 SAAM Bash Console 60  
 SAAM Command line info 187  
 SAAM extranet 181  
 SAAM Intuitive Maker (SIM) 190  
 SAAM Presentation 256  
 SAAM Reference Manual 187  
 SAAM to Unix Data Flow 181  
 SAAM Training Courses 187  
 SAAM Viewport 19  
 SAAM what's new 187  
 SAAM window 19  
 SAAM\_default\_scenario 16  
 SAAM\_USER\_PREF 16  
 Save 24  
 Save As 24  
 Scenario 150  
 Scenario Dashboard 190  
 Scenario Economy 127  
 Scenario from Bretigny's Real Time 181  
 Scenario Processing 187  
 scheme 293  
 script files 60  
 Scrolling through graphs 228  
 Sea Colour 190  
 search criteria 66  
 Search Criteria Suffix 73  
 Search for Airblock(s)/Sector(s) 196  
 Sector file 17  
 Sector Workload (Macroscopic formula) 112  
 Sector Workload / Capacity (CAPAN like) 116  
 Sector\_SLS to Center\_SLS 164  
 Sectors 196  
 Sectors to bypass 39  
 Segment Load Comparison 106  
 Select Airblock/Sector 196  
 Series properties 229  
 Set as Default Scenario 24  
 Set Level View Ref to ground 186  
 Set Side view 186  
 Set Top view 186  
 shell 360  
 shell commands 60  
 shell console 60

SHER 94  
 Side view 186  
 SIDs and STARs 241  
 SIM 56  
 Sliding Hourly Entry Rate 94  
 SLS file 17  
 Small Video Training 187  
 So6 To Direct Route PIN 149  
 So6 to List of Options 147  
 So6 to PIN 148  
 SO6 to Traffic Demand 145  
 Solid 196  
 Split Airblocks 196  
 STARs 241  
 Start-up Icon 16  
 STATFOR 51  
 status bar 19  
 Stereo Mode 190  
 sub-presses flights 70  
 sub-SO6 File 71  
 Supervision thresholds 116  
 Surface / Volume 85  
 Swap 2D/3D 190

## - T -

T5 Cleaning Rules 46  
 TAAM 172  
 TDV Editor 239  
 TDV file 190  
 Text Features 216  
 Tilt 21  
 title bar 19  
 Toggle lighting for surfaces 190  
 Toggle show negative volume 190  
 Toggle smooth shading 190  
 Tool Bar 19, 190  
 Top view 186  
 Traffic data 17  
 Traffic Demand 28, 241  
 Traffic Intersection file 46  
 Traffic Punching 158  
 Training 15  
 Transform 144  
 Transparent 196  
 type of segment 216

## - U -

Undo 27  
 Ungroup 196

User Support 15  
 Using Airports or Zones 241

## - V -

vertically 21  
 Via Rules 241  
 Video Size 186  
 Video Training 187  
 View 186  
 View Mode 196, 216  
 viewport 19  
 Visualisation 10  
 VST network 17

## - W -

Where-am-I 27  
 Window 19  
 Workload calculation 114

## - Z -

ZIN 241  
 ZIN file 28  
 Zoom In 21  
 Zoom Out 21

