**The SAS CMS Project**

**Functional Reference Manual**

*Carmen Crew Rostering*

*Document Type: Functional Reference*

*Version: 1.31*

*Last Changed: 03 Jul 2018*

Table of Contents

[1 Introduction 7](#_Toc503355556)

[2 The plan concept in CMS 8](#_Toc503355557)

[2.1 Database plan 8](#_Toc503355558)

[2.2 File Plan 8](#_Toc503355559)

[2.3 How-to 9](#_Toc503355560)

[3 Planning Area 13](#_Toc503355561)

[3.1 Predefined areas 13](#_Toc503355562)

[3.2 Rave 13](#_Toc503355563)

[3.3 Dave 13](#_Toc503355564)

[3.4 Filters 13](#_Toc503355565)

[4 Fairness 15](#_Toc503355566)

[4.1 Introduction 15](#_Toc503355567)

[4.2 Concept 15](#_Toc503355568)

[4.3 The Fairness package 15](#_Toc503355569)

[4.4 How does fairness work? 16](#_Toc503355570)

[4.5 User configuration 18](#_Toc503355571)

[4.6 System configuration 19](#_Toc503355572)

[4.7 Usage 21](#_Toc503355573)

[4.8 Fairness Crew Groups 21](#_Toc503355574)

[4.9 Fairness Elements 22](#_Toc503355575)

[4.10 Reports 24](#_Toc503355576)

[5 Preferential Bidding System – PBS 25](#_Toc503355577)

[5.1 Introduction 25](#_Toc503355578)

[5.2 Lifestyles 25](#_Toc503355580)

[5.3 User configuration 29](#_Toc503355581)

[5.4 System configuration 30](#_Toc503355582)

[5.5 Usage 30](#_Toc503355583)

[5.6 Bid types 33](#_Toc503355584)

[5.7 Visual indicators 35](#_Toc503355585)

[5.8 Planning tools 36](#_Toc503355586)

[6 Optimizer Cost function 38](#_Toc503355587)

[6.1 Planning period 38](#_Toc503355588)

[6.2 Parameters 38](#_Toc503355589)

[6.3 Cost elements 38](#_Toc503355590)

[6.4 Constraints 53](#_Toc503355591)

[7 Compensation days 57](#_Toc503355592)

[7.1 Parameters 57](#_Toc503355593)

[7.2 Filters 57](#_Toc503355594)

[7.3 Assign function 57](#_Toc503355595)

[8 Assignment functions 58](#_Toc503355596)

[8.1 Standby 58](#_Toc503355597)

[8.2 Blank days 62](#_Toc503355598)

[8.3 Compensation days (bid) 63](#_Toc503355599)

[8.4 Freedays 64](#_Toc503355600)

[8.5 Special Schedules 64](#_Toc503355601)

[8.6 Crew Warnings 66](#_Toc503355602)

[8.7 Rule Exceptions 69](#_Toc503355603)

[9 GUI 73](#_Toc503355604)

[9.1 Filters 73](#_Toc503355605)

[9.2 Filter variables 76](#_Toc503355606)

[9.3 Sorts 77](#_Toc503355607)

[9.4 Find 78](#_Toc503355608)

[9.5 Temporary filtering 78](#_Toc503355609)

[9.6 Predefined filters 78](#_Toc503355610)

[10 Overlapping Activities 80](#_Toc503355611)

[10.1 Resolving overlaps 80](#_Toc503355612)

[11 Publish 82](#_Toc503355613)

[11.1 Basic publishing step 82](#_Toc503355614)

[11.2 Crew changing planning area in the publication period 84](#_Toc503355615)

[11.3 Conflict during Publish 85](#_Toc503355616)

[11.4 Republish 85](#_Toc503355617)

[12 Working with Database 86](#_Toc503355618)

[12.1 Planning period and area 86](#_Toc503355619)

[12.2 Database, Studio, Application synch 86](#_Toc503355620)

[12.3 Saving and refreshing 87](#_Toc503355621)

[13 Fly Below Rank 88](#_Toc503355622)

[13.1 Flight Deck 88](#_Toc503355623)

[13.2 Cabin Crew 88](#_Toc503355624)

[13.3 Reassignment script 88](#_Toc503355625)

[14 Max production days per vacation year 89](#_Toc503355626)

[14.1 Background 89](#_Toc503355627)

[14.2 Rules 89](#_Toc503355628)

[14.3 Change rule calculation dates 89](#_Toc503355629)

[14.4 Temporary service grade 90](#_Toc503355630)

[14.5 Exclude crew 91](#_Toc503355631)

[14.6 Report 91](#_Toc503355632)

Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Description |
| 0.1 | David Lennartsson | 2006-01-01 | Initial document |
| 0.2 | David Lennartsson | 2006-11-20 | Tx0 and Tx1 additions |
| 0.3 | Erik Gustafsson | 2006-11-07 | Optimizer cost section. |
| 0.4 | Martin Karlsson | 2006-11-27 | Final adjustments for TX1 delivery |
| 0.6 | Jonas Carlsson | 2007-01-22 | Tx2 additions on PBS |
| 0.7 | Erik Gustafsson | 2007-03-16 | Assignment functions section added. |
| 0.8 | Jonas Carlsson | 2007-03-19 | PBS Section updated, Standby and blank day assignment sections added |
| 0.9 | Erik Gustafsson | 2007-03-23 | Updated after TX3-release (separated GUI to own section, updated Special schedule section) |
| 0.10 | Erik Gustafsson | 2007-04-02 | Updated Flight attributes and Fixed freedays section. |
| 0.11 | Jonas Carlsson | 2007-04-27 | Updated section on PBS bid type interpretation and roster bid carry out period |
| 0.12 | Erik Gustafsson | 2007-05-02 | Updated Flight Attributes section. |
| 0.13 | Jonas Carlsson | 2007-05-14 | Updated cost function section with new stability and quality costs and vertical constraints. |
| 0.14 | Jonas Carlsson | 2007-05-16 | Added section on Crew Warnings |
| 0.15 | Jonas Carlsson | 2007-05-18 | Added freeday monthly target stability cost |
| 0.16 | Jonas Carlsson | 2007-05-18 | Updated section on standby optimization and blank days |
| 0.17 | Erik Gustafsson | 2007-05-18 | Corrected a few spelling mistakes, changed some headers to correct the TOC, added section on post-optimization freedays, added all selects and sorts to separate section. |
| 0.18 | Jonas Carlsson | 2007-05-21 | Updated section on crew warnings and minor corrections on standby optimization. Added section on Rule Exceptions |
| 0.19 | Jonas Carlsson | 2007-07-09 | Updated info on BidConfig table in PBS chapter. |
| 0.20 | Anna Olsson | 2007-08-13 | Added short descriptions of select filters |
| 0.22 | Jonas Carlsson | 2007-08-20 | Updated crew warning section |
| 0.23 | David Hellerström | 2007-08-20 | Added training, duty-codes and database-plan information |
| 0.24 | ? |  |  |
| 0.25 | Anna Olsson | 2007-08-29 | Added some new trip window selects |
| 0.26 | David Hellerström | 2007-09-14 | Added more information on Crew training Log and training assignment |
| 0.29 | Jonas Carlsson | 2007-10-03 | Added information on PBS and handling of bid and bid rubob files |
| 0.30 | Anna Olsson | 2007-10-23 | Updated definition of select Nop leg incl. other fleet. |
| 0.31 | Jonas Carlsson | 2007-10-30 | Updated section on Fairness, elements and crew groups |
| 0.32 | Anna Olsson | 2007-11-21 | Added costs for part time summer vacation and New flights |
| 0.33 | Erik Gustafsson | 2007-11-27 | Added Recurrent Training Optimization. |
| 0.34 | Anna Olsson | 2007-12-07 | Added additional info about database plans to rule exception section. |
| 0.35 | Anna Olsson | 2007-12-14 | Added section about select variables |
| 0.36 | Jonas Carlsson | 2008-01-24 | Created a v.36 from alpha1\_v2 |
| 0.37 | Jonas Carlsson | 2008-02-04 | Replaced a reference to CTL with Crew Training |
| 0.38 | Erik Gustafsson | 2008-02-05 | Corrected sections on training cost elements, constraints and the section on training. Removed old screenshots of CrewTraining, separate documentation should be used. |
| 0.39 | Anna Olsson | 2008-02-05 | Added costs for part time summer vacation |
| 0.40 | Erik Gustafsson | 2008-02-18 | Added info on flight attributes for simulators. |
| 0.41 | Jonas Carlsson | 2008-02-19 | Removed old text on fairness crew groups |
| 0.42 | Jonas Carlsson | 2008-03-03 | Merged in updates in the Alpha5 documentation |
| 0.43 | Jonas Carlsson | 2008-03-03 | Added documentation on PGT Allotment. Added crew warning U for Flight Deck. Added more info on overlapping activities commands. |
| 0.44 | Jonas Carlsson | 2008-03-12 | Added more documentation on PGT Allotment. Cleaned up the usage of crew category names and the use of Reserve vs. Standby. Cleaned up the use of FD/FC/CC/FlightDeck/CabinCrew |
| 0.45 | Jonas Carlsson | 2008-03-13 | Minor update on reference roster level calculation. Added documentation on temporary selections. |
| 0.46 | Jonas Carlsson | 2008-03-13 | Updates on the load plan dialog |
| 0.47 | Jonas Carlsson | 2008-03-14 | Minor update to Publish, and some clarifications on PBS. |
| 0.48 | Erik Gustafsson | 2008-04-21 | Added info on database. |
| 0.49 | Per Grönberg | 2008-04-29 | Added info in assign legal freedays and changes to publish. |
| 0.50 | Per Grönberg | 2008-05-13 | Added in to special schedule |
| 0.51 | Erik Gustafsson | 2008-06-10 | Some fixes related to planning areas |
| 0.52 | Per Grönberg | 2008-06-12 | Added info in fairness about automatic copy of fairness config |
| 0.53 | Erik Gustafsson | 2008-06-13 | Added FBR section and updated fairness usage. |
| 0.54 | Erik Gustafsson | 2008-08-22 | Minor FBR fix. |
| 0.55 | Per Grönberg | 2008-09-03 | Assigning compday/preop/postop |
| 0.56 | Per Grönberg | 2008-09-09 | Updated publish with holes/overlap-note and crew info-special schedule |
| 0.57 | Erik Gustafsson | 2008-09-26 | Removed training information |
| 1.0 | Erik Gustafsson | 2008-10-06 | Final adjustments |
| 1.1 | Erik Gustafsson | 2008-10-23 | Added BL reduction to costs. |
| 1.2 | Per Grönberg | 2008-10-24 | Updated open db-plan picture |
| 1.3 | Erik Gustafsson | 2008-11-17 | Fixes for SP1 |
| 1.4 | Per Grönberg | 2008-12-03 | Special weekend assignment |
| 1.5 | Erik Gustafsson | 2009-01-09 | Added rule for FBR assignment to certain quals. |
| 1.6 | Janne Carlsson | 2009-01-16 | Note added for sort SBY number in year |
| 1.7 | Erik Gustafsson | 2009-03-18 | Added reward for troublesome rank |
| 1.8 | Janne Carlsson | 2009-03-30 | Changes in absence crew warnings |
| 1.9 | David Karlsson | 2009-04-08 | Changes in Assign Post-op Freedays. |
| 1.10 | Erik Gustafsson | 2009-04-15 | Updated cost 3.3 |
| 1.11 | David Karlsson | 2009-04-17 | Added cost element for 2 pilot flights. |
| 1.12 | Erik Arnström | 2010-02-08 | Updated cost 3.3 |
| 1.13 | Erik Arnström | 2010-06-29 | Misc. corrections |
| 1.14 | Pierre Nordblom | 2010-08-25 | Updated cost 2.6 and 2.7 |
| 1.15 | Erik Arnström | 2010-09-10 | Added FBR reassignment (CR 348) |
| 1.16 | Anna Olsson | 2011-01-27 | Added constraint for limitation resource pool crew on flight. |
| 1.17 | Anna Olsson | 2011-06-15 | Added new fairness elements |
| 1.18 | Arvid Berzelius | 2012-09-21 | Added documentation for max production days per vacation year |
| 1.19 | Anton Hedin | 2012-09-28 | Added documentation for long term fairness |
| 1.20 | Niklas Johansson | 2012-10-29 | Updated standby-line optimization |
| 1.21 | Arvid Berzelius | 2012-10-29 | Changed menu entries for Select to Filter |
| 1.3 | Viktor Almvist | 2013-02-26 | CMS2 updates. |
| 1.31 | Emil Rinaldo | 2018-07-03 | Lifestyle updates. |

# Introduction

This document, the functional reference for Carmen Crew Rostering, is describing all functionality that is customized for rostering in the SAS CMS project.

Reports, rules and administrative tasks are described in separate documents. Also functionality shared between other parts of CMS will be documented separately. Although the processes around certain areas of functionality are touched upon, the complete workflow will be described in a Step-by-Step guide handled by PG.

# The plan concept in CMS

## Database plan

The most important plan in the CMS system is the production database plan. The plan is a continuous roster where all departments active in the crew planning- and follow-up-process works against the same common production plan. However, different departments are responsible for writing in different parts of the roster. There are different applications working against the same plan. Such applications are Studio (for Tracking, Planning or PreRostering), Manpower and stand-alone applications such as Crew Info. Database plans enable multi-user access to one common plan so that the whole organization shares the same world-view.

## File Plan

As a complement to the database-plan planners have the possibility to create file-based plans. The file-based plans are similar to what is used in today’s pairing and rostering installations. These plans are used for rostering and pairing scenarios, optimization and testing.

File-based plans can either be created from SSIM or CTF files or created by exporting a rostering scenario from a database plan. When working against file-based plans the database-coupled ModelServer is not available to the user. Therefore WAVE-applications (such as Crew Info, Crew Accounts) that rely on the database are inaccessible while connected to a file-based plan.

Access to file-based plans is restricted to one single user. If a second user attempts to access the same plan, a dialog is presented with a warning that the plan is locked and access will be read-only.

The possibility of multi-user editing and online updates available in database mode makes it desirable to work as much as possible against a database. Figure [1] illustrates an example of a planning-process divided between database-plan and scenarios are presented.



1. Example of the planning process, divided between file and database storage.

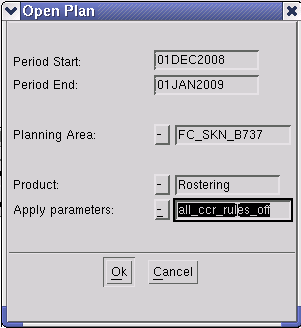
## How-to

A number of important steps in the CMS planning process when it comes to database interaction are described in this section.

### Open a Database Plan

A database plan is opened from the plan manager just like a file plan. When the plan is opened a load filter dialog appears where the period, planning area, product and parameter set can be selected before fetching data from the database. Product options are Rostering and Pairing.

The database-plan names are written in lower-case letters and are in the test environment called things like Deliveries/Alpha/a1\_static/a1\_static.



1. Open database plan dialog.

#### Period

The actual period opened from the database is the specified planning period and an additional calendar month before and after the planning period.

**Example:** If the user enters start of planning period = 1Feb2008 and end of planning period = 1Mar2008 then the period opened from the database will be 1Jan2008-1Apr2008. The planning period in Rave will be set to 1Feb2008-1Mar2008.

#### Planning Area

The planning areas that can be selected are predefined. More about Planning area in section 3.

#### Product

CMS behaves a bit differently for Pairing and Rostering. Choosing Rostering will ensure they right functionality for rule sets, export etc.

#### Parameter Set

The parameter sets that can be chosen are those that are saved by the user in $CARMDATA/RaveParameters/<user>. When clicking OK the selected data is loaded from the database. If no parameter set is chosen the default parameters of the rule set will be applied.

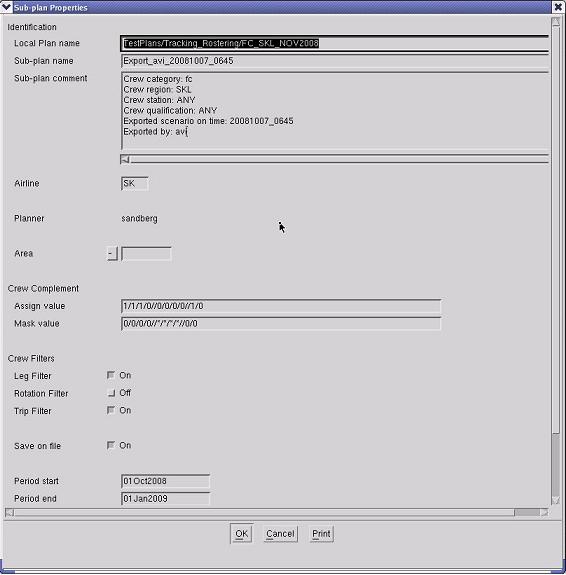
It is possible at any time to load, or save, a parameter set from the parameter form.

#### Rule Set

When plan is opened from the database Studio loads an appropriate rule-set, Rostering\_CC for cabin areas and Rostering\_FC for flight-deck areas.

### Export Scenario

Rostering optimization is done in a file plan mode only! For that, a rostering scenario is exported using; **Select File -> Export -> Scenario** from the File menu. A confirm export dialog is opened. When OK is chosen, a new dialog is opened where sub-plan properties may be set before saving (see Figure [3]). The default local and sub plan names are based on the data chosen when opening the database, together with a timestamp and username.



1. The sub-plan properties are set in this dialog before exporting a rostering scenario.

When OK is clicked the scenario is exported and a confirmation message stating “ExportScenario::Finished ok, you are now in file mode” will show.

This operation will also export a snapshot of all etables used by planning and stores these in LpLocal (local plan external tables) and SpLocal (sub-plan external tables) in the plan. Tables in LpLocal are shared between all sub-plans in the same local-plan. Only tables usually changed in sub-plans are stored in the SpLocal. The user can override this by copy a file from LpLocal to SpLocal. In that case the file in SpLocal will be used instead.

### Fetch assignments

When production quality rosters have been built in a file scenario, using the rostering optimizer, it is time to save these rosters in the database. This is done using the Fetch Assignments functionality.

A few typical steps included in this operation are:

1. Open the database and load the appropriate planning area
2. Lock everything that should be left on the roster and is not yet locked, to make sure it is not deassigned by mistake.
3. De-assign everything that should be replaced by the assignment solution to be fetched
4. Fetch Assignments from file based solution
   1. **Planning Tools -> Subplan -> Fetch -> Assignments**…
      1. Local plan
      2. Sub plan
      3. Optimizer solution
   2. Set parameters:
      1. “Match trips in current subplan”
      2. “Deassign trips in current subplan”

These options will make sure that if there is space on the roster in the database, an assigned trip from the file-plan will be put there, otherwise it will “fall off” and end up in open time. Trips from the file-plan will be matched against the trips in the database, this means that trips that fall off does not create new trips and an assignment consumes a position on an existing trip. If no match is found the trip-assignment will not be fetched. This is the desired behaviour because the database should contain all production trips and thus no new trips should generally be created in the rostering step.

Also, the only assignments of interest to fetch are probably flight duties, standbys and blank days in the planning period. This can be specified using the Rave filter values in the fetch dialog.

1. Repair the rosters. It might be the case that assignments fall off since there is something else assigned on the database roster. The fetch log will give information on this, and help the planner identify if it is something that needs to be taken care of.
2. The last step is to save the changes. If everything went ok, and the fetched assignments are correct, it is time to save. It is a good idea to enter a descriptive text in the save comment dialog, stating what was fetched and why. If the fetch was not ok, or for some reason it is not desired to save, then close the database without saving, and then re-open.

For more information about the Fetch operation please check the Carmen Crew Rostering online help.

# Planning Area

Planning areas are used to divide the complete SAS planning problem in different areas. Planning area defines which crew are used to man which flights and which planning department within SAS that is responsible for creating the trips and the rosters.

The planning area filters available to the planners are active in two ways; when working within Studio and when opening a database plan.

## Predefined areas

For all regular planning areas there should be a pre-defined option available already when opening the plan (see section [2.3.1]), this should be enough to setup the planning.

However there are possibilities to slightly change this after loading. Changes to the crew area will not have much effect since crew is filtered already on load, for trips and legs the area is completely defined in Rave and thus any change to the area after load will have an effect.

## Rave

To define a planning area a number of Rave parameters must be set using the Rule Parameter form, tab Period and Area. There is a section for Crew, Trip and Leg where the details of an area can be defined. In the info window in the lower right corner the current planning area settings are displayed.

After configuring the planning area these settings will be a part of various filters, in particular all filters from the **Assignment/Trip/Leg General -> Filter**.

The Rave filters are sensitive to the planning period; they will change if the planning period is changed. This mostly affects the crew filtering since the crew properties that are used for filtering only are valid a certain time period.

## Dave

When opening a database plan the user is prompted for a planning area, in the list all the normal planning areas plus some other filters are available. Apart from configuring the Rave filter it also affect what data is loaded into Studio.

The planning area filtering Dave does not affect what legs are included in the load from the database, all legs are loaded and the Rave filters will have to be used in Studio. For crew and trips however, only those which matches the planning area at some point in time during the opened interval will be loaded. The Rave filter will then be applied within the planning period which means that there might be more crew and trips loaded from the database than are in the planning area in the specified planning period.

It is recommended to use as specific a planning area as possible in order to reduce load time and memory consumption in Studio

## Filters

Crew, unassigned trips and legs are affected by planning area filters.

### Crew

The planning area for crew is made up of Category, Region and Qualification. Region is used to determine the Agreement used in legality, and since that matches the planning department responsible for planning crew, Region is also used as the planning area selection.

For the Rave filters it is possible to set a specific Station but it is not recommended, leaving it set to Any should be enough.

### Trip

The trip filter is made up from Homebase, Region (defined using Homebase), Category the trip is for and the Aircraft families of the legs in the trip.

The trip filter does not only look at active flight trips but some ground duties can also be caught by the trip filter. Standbys with qualification codes and simulators (which have the same codes) will all be filtered using the Aircraft family property of the filter.

If a trip has an explicit area set, this will override any other values. See Functional Reference Common for more info.

### Leg

Legs are filtered on Aircraft family and Region (cabin\_crew\_employer).

# Fairness

## Introduction

This chapter describes the Fairness functionality implemented for SAS. Some of the text below is copied from the documentation for the Standard Fairness-package. Before developing your own fairness elements or administrating the fairness implementation it is recommended to consult that document. This document focuses on the functions enabled in the SAS installation.

## Concept

The basic concepts of fairness are illustrated using the following example:

Suppose that there is only a small crew of three captains, two working full time and one working 50%. If their block hours during a month are randomly allocated it might look like Figure [4].



1. A fair distribution that considers the work rate is 60 hours for the full time crew and 30 hours for the part timer.

* The total work to be done is 65 + 45 + 40 = 150 hours.
* The total work rate is 100 + 100 + 50 = 250%.
* The 100% target is 150 / 2.5 = 60 hours.
* The 50% target is 60 x 0.5 = 30 hours.

The fairness functionality aims to minimize the differences between crew. It is not certain that they should have the same amount of hours; somebody with a lower work rate should also have a lower target. Each captain's personal target can be calculated by multiplying the full time target with the personal work rate.

To get a fair distribution we try to minimize the deviation between the actual value and the target for all crew. The deviations in the example above are +5h, -15h and +10h. The average deviation is (5 + 15 + 10) / 3 = 10 hours. A low average deviation indicates that the fairness distribution is good, but does not tell anything about whether there are some extreme cases or if everyone has the average deviation.

## The Fairness package

The Carmen fairness package is normally used to achieve the even distribution described above, but it is also possible to let the target be influenced by seniority or letting different groups of the crew have different targets.

The generic code suits different kinds of fairness aspects, which in the following text are called fairness elements. Each fairness element needs customized rave code to calculate the monthly work and roster values. New elements can easily be added by writing a few lines of rave code.

The main purpose of the package is to calculate a cost of the target deviation for each element for all crew. The cost for all fairness elements is included in the global cost function, used by the optimizer, and blended with other cost components like open trips, preferences, quality and stability. Some costs may be in conflict with each other. For example preferences are about giving everyone what they want and fairness is about giving everyone the same. Therefore the tuning requires a good understanding of the whole rostering problem.

The settings for the calculations are defined in *etables*. An initial tuning is typically done by an implementation project from Carmen, but when the rostering changes these settings needs to be reviewed. This document describes how to configure the elements from a user’s perspective.

The fairness package is not designed to handle change of crew functions within the planning month, neither as a promotion nor as working down temporarily. The crew function in the beginning of the planning month is used for the whole month. If the amount of work is similar for all functions it has a minor impact, if not the target will be wrong for crew that are changing.

## How does fairness work?

### Fairness crew group

The fairness crew group defines what crew and trips are grouped together so that work can be evenly distributed within the group. It can be partitioned by for example rank and base in a planning area where there are no base-variants and no fly below rank. If fly-below-rank is utilized, it might be good to share work between ranks. One way to accomplish that is to only utilize base as definition of fairness crew group. Despite being called crew group trips must be partitioned with the same key.

The fairness crew group is defined within each planning problem.

### Work rate

The work rate is the amount of work in full time equivalents a crew member shall pick-up. Work rate is calculated as a product of contract work rate and roster work rate. The roster work rate is a reduction due to non-freeday off-duty activities (illness, leave of absence, vacation) or pre-assigned on-duty activities not contributing to the fairness element. An example of the latter case might be that an on-duty simulator that does not contribute with block-time shall lower the block-time target, while it should not lower the duty-time target.

### Work

The total amount of the element in the plan. This is work pre-assigned on the crew and left in open production before the optimization. Since work rates are pre-calculated objects assigned by the optimizer can’t change the work-rate. Therefore it is important that all objects assigned by the optimizer, that should change the amount of work crew shall do, contribute with some standard value. For example, a standby might not contribute with duty-time legality-wise, but fairness-wise it is good if it contributes with an amount similar to a duty-day. Otherwise a crew, which the optimizer assigns a standby to, will only receive long duties to achieve his fairness goal that month.

### Calculation of targets

The target for a full time equivalent (F.T.E) is calculated as

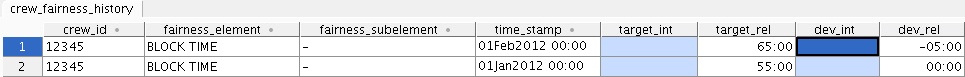
***Total Work / Total Work Rate***

The target for a specific crew is calculated as

***Personal Work Rate x F.T.E target***

### Long term fairness

A fairness element can use and keep track of the accumulated deviation by enabling IncludeInHistory in the table *fairness\_admin*. This is used for long-term fairness. The deviation is stored in the database table *crew\_fairness\_history*. When adding a new element the internal history will not exist and by default it will be counted as zero.



1. *crew\_fairness\_history* table

Elements are identified in the columns *fairness\_element* and *fairness\_subelement* and the values are given in the following columns:

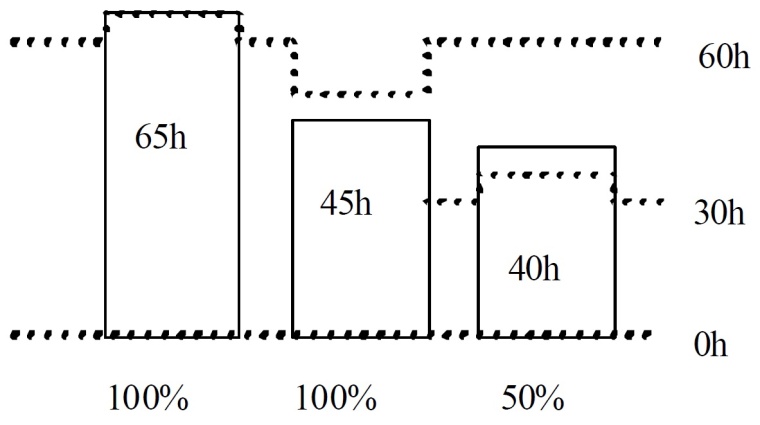
*time\_stamp* indicates the month the row belongs to (always the 1st of the month considered).

*target\_int* (or *target\_rel* depending on the type of the fairness element) stores the personal target, calculated in file plan, for the crew.

*dev\_int* (or *dev\_rel*) stores the accumulated deviation from target, which is used to adjust each crew member’s personal target in the next planning period.

When exporting a scenario to file, the *crew\_fairness\_history* table is exported as well, and the deviation values are used by rave to adjust the personal targets for fairness elements which have been enabled for long term fairness in the table *fairness\_admin*.

Consider again the example with block hours from section 4.2. If the block hours should be distributed on an annual basis instead we have a history that must be considered, the accumulated deviation from previous months in the year. In the example below the crew have the same work rates and hours for this month, but the accumulated deviation is -5h, +10h and -5h.



1. When the history is considered the new target for this month will be calculated as ***personal target – accumulated deviation*** which is 65, 50 and 35.

When fetching assignments from file plan to data base plan the personal targets for the crew in the file plan (if they have been calculated) will be stored in *crew\_fairness\_history*. At roster publish the personal target is compared to the roster value to calculate the deviation

***Dev = Roster Value – Personal Target***

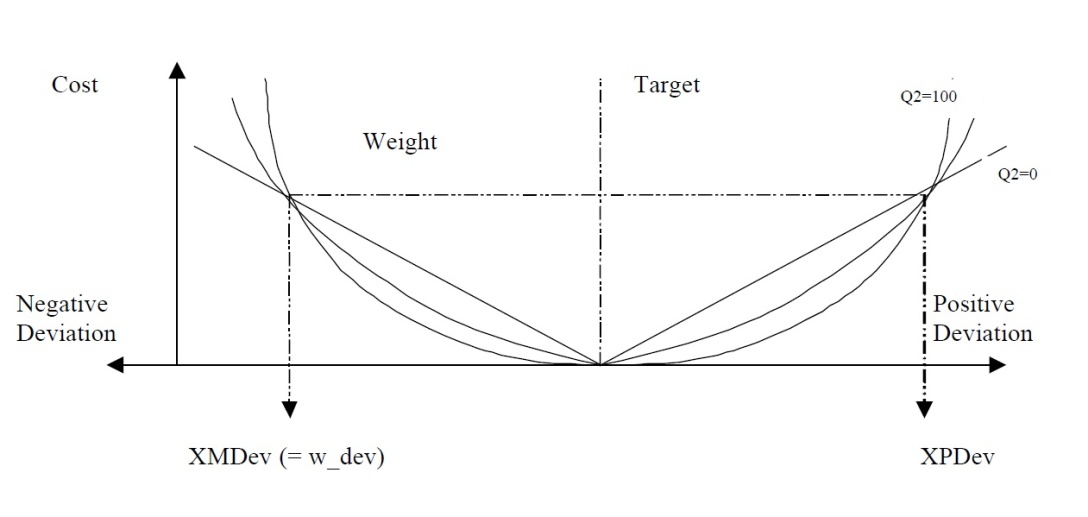
which is then stored in *dev\_rel* (or *dev\_int*). In the block hours example the crew would then have 0h, -5h and +5h as their deviations to be used as accumulated deviations in the next month.

### The cost function

The cost function is used to penalize deviations from the target value. A larger deviation should give a higher penalty. Elements that are measured only will not need any cost function.

The basic idea of the cost function is: At the deviation *w\_dev* from the target, the penalty is equal to *weight* from the *fairness\_config* table. This works as a normaliser to the cost function, *w\_dev* can be set differently for negative and positive deviations, *XMDev* and *XPdev*. A zero value for *XMDev* or *XPDev* indicates that there is no cost for negative/positive deviations. This can be useful if the penalty only applies when being below or above the target. The cost function is based on one linear part and one quadratic part. The linear part is there to make sure that the penalty doesn't grow too fast. The Q2 factor defines how quadratic the function is, 0 for a linear function up to 1.00 for a completely quadratic function.

See Section 4.6.1 for how to set these values.



1. Fairness cost function.

### Process

* Initial calculation of work rates and targets. This is needed for the reports to work.
* Manual rostering and pre-assignments
* Re-calculation of work rates and targets to consider changed work rates and amount of production
* Optimization
* Post optimization analysis with reports

## User configuration

Fairness elements are controlled and activated in the table *fairness\_config* (see Table [1]). This table controls the currently activated fairness elements. There can be more elements defined in Rave and in the administration tables, but they will be ignored in reports and optimization.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| seqno | string1 | string2 | sendev | maxovershoot | weight | distribute |
| 1 | “DUTY TIME” | “” | 0 | 300 | 10 | true |
| 2 | “STANDBY” | “” | 0 | 0 | 20 | false |
| 3 | “BLOCK TIME” | “” | 0 | 0 | 10 | true |
| 4 | “CC DUTY TIME” | “SH” | 0 | 0 | 0 | false |
| 5 | “CC DUTY TIME” | “LH” | 0 | 0 | 0 | false |
| 6 | “CC DUTY TIME” | “TOT” | 0 | 0 | 20 | false |
| 7 | “CC DUTY DAYS” | “SH” | 0 | 0 | 0 | false |
| 8 | “CC DUTY DAYS” | “LH” | 0 | 0 | 5 | false |
| 9 | “CC DUTY DAYS” | “TOT” | 0 | 0 | 10 | false |

1. fairness\_config

The *seqno* is used internally to loop over all elements and must be between 1 and 40. There cannot be more than 40 elements at the same time. An element can be turned off temporarily by setting *distribute* to false. It is the same as removing the line from this table. The elements are identified by *string1* and *string2*. These strings are used as keys when looking up element attributes in all the other tables defining the fairness element.

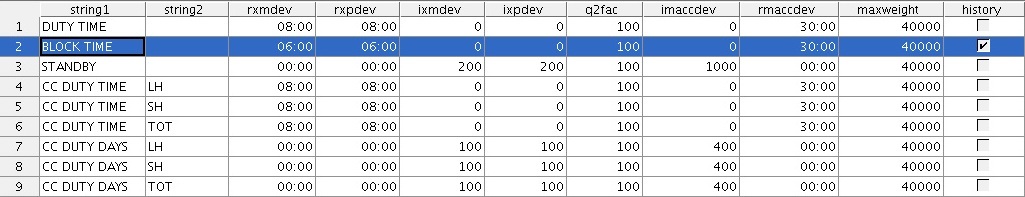
The *weight* defines the importance of a fairness element. The deviation penalty will be multiplied by this value before being added to the cost. Note that elements that are measured, but not controlled should have the weight set to 0. Those elements are still included in all reports but will result in zero cost in optimization.

The s*endev* can be used to get seniority dependent work rates. Seniority deviation for fairness is currently not in use at SAS.

The *maxovershoot* value defines a hard limit to how much a fairness element can exceed the target on a crew without making it illegal. It is only used for performance since it restricts the optimizer to build completely unfair rosters. Element values are in minutes for elements of type RelTime and in fractions-of-hundred when the type is Integer. The types are defined in the Rave-code.

## System configuration

System configuration is done in the table *fairness\_ admin* (together with code-changes in rave and python).



1. *fairness\_admin* table

Elements are identified by *string1* and *string2*, and values by:

*rxmdev* - point where cost equals weight (from *fairness\_config*) for negative deviations (reltime)

*rxpdev* - weight point for positive deviations (reltime)

*ixmdev* - weight point for negative deviations (integer)

*ixpdev* - weight point for positive deviations (integer)

*q2fac -* An integer. 0 means completely linear behaviour, 100 means completely quadratic behaviour

*(imaccdev* - Maximum accumulated (history) deviation for integer, including two decimals. Currently not used.)

*(rmaccdev* - Maximum accumulated (history) deviation for reltime. Currently not used.)

*maxweight* - Maximum cost for this element

*history* - true if the element uses should be considered for long term fairness.

Depending on whether the element is a reltime or an integer element, different columns are used.

The *maxweight* defines the maximum cost for an element. Larger costs will be truncated. Without any limitations on the cost there can be a risk that one fairness element has a too large part of the total rostering optimizer cost. On the other hand a low *maxweight* can cause problems because crew that has the maximum cost can get even larger deviations without any increased penalty. The effect could be that a bad roster gets beyond salvation and is sacrificed to improve other rosters!

### Setting the values

The following guidelines are meant as a help in setting the values for a fairness element (using the tables *fairness\_admin* and *fairness\_config*). A reminder: ALL integer values, in external tables and elsewhere, are expressed with two decimals and are 100 times larger than they really are to make it possible to use the RAVE data type integer. For instance 150 means avalue of 1.50 and 8 means a value of 0.08.

You have to set the following values:

***rxmdev* and *rxpdev* in *fairness\_admin* table**(Reltime; on the X-axis, the Plus/Minus Deviation)

This is the "deviation point" with an associated 'Weight', see below. This is the only non-trivial value to be decided upon. The deviation point shall be set so that 20 deviations from the target are never reached. Any smaller number is good, use 10 deviations as a starting attempt.

Example: If the fairness element is Block Time in a Month, the target might be around 50 hours. If we judge that 0-100h are plausible values in a roster, 5h could then be an appropriate deviation, because 50 hours plus/minus 10 deviations of 5 hours each means a range between 0h and 100h.

***ixmdev and ixpdev* in *fairness\_admin* table**   
(Int; on the X-axis, the Plus/Minus Deviation)

The deviation point as an Integer can be typically set to 1. Examples include distribution of number of layovers and number of extra freedays.

***maxweight* in *fairness*\_*admin* table**

Reasonable values are in the interval 10000 - 100000. Use 40000 as the normal weight for a fairness element. If later you would like to state that one fairness element is more important than another, increase this value.

***q2fac* in *fairness\_admin* table =** 100

This gives a quadratic penalty.

## Usage

### General process

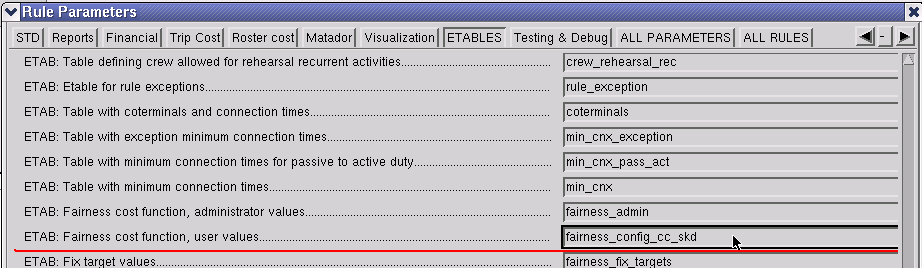
1. If desired, set up planning area specific configuration as described in 4.7.2
2. Export a rostering scenario and open it
3. If step 1 hasn’t been performed; open the Table Editor for File and copy the *fairness\_config* table from Config tables to Subplan External Tables.
4. If needed, make additional changes in the *fairness\_config* table in Subplan External Tables
5. Set fairness crew group parameters
6. Calculate work rates and targets (**Planning Tools-> Create Tables-> Fairness targets**).
7. Set Optimizer cost function parameters and run optimizer
8. Post optimization analysis with reports
9. Fetch personal fairness targets for long term fairness. (This is automatically done when fetching assignments to the data base).

### Planning Area specific configuration

Each planning area will typically have its own setup for fairness. This is supported by having multiple fairness\_config etabs in $CARMDATA/ETABLES, e.g. $CARMDATA/ETABLES/fairness\_config\_cc\_skd.etab.

To use this feature:

* + Copy the standard *fairness\_config* table from Config Tables to Data Tables in the **Table Editor-> File** and give it a unique name.
  + Edit the copied table for the specific planning area.
  + Set the corresponding parameter to the new table and save the parameter set for future use (see Figure [9]).



1. Setting the corresponding parameters in ETABLES.

Without all the steps described above the system will use the standard config table.

## Fairness Crew Groups

The fairness crew groups in the SAS installation have been hard coded, due to how the planning areas are set up and the planning process.

For flight deck all crew are within the same group, neither base nor rank are used. This is because the heavy use of base variants and fly-below-rank in practice means that all crew share all work, and a partitioning by rank would make the FP target very unrealistic since there are many more FC.

For cabin crew the crew groups are partitioned by rank and if crew have short-haul or long-haul qualifications (see [4.9.4]).

## Fairness Elements

### Duty time

**Element:** DUTY TIME

**Subelement:** N/A

The fairness element tries to distribute the number of on-duty minutes for crew.

The work rate for DUTY TIME is reduced by non-freeday activities that don’t contribute to the DUTY TIME element. Pre-assigned freedays are transparent to the work rate calculation. Freedays are taken into account by the contract work rate.

### Standby

**Element:** STANDBY

**Subelement:** N/A

The fairness element tries to distribute the number of standby (=reserve) days for crew.

The work rate for STANDBY is reduced by non-freeday activities that don’t contribute to the STANDBY element. Pre-assigned freedays are transparent to the work rate calculation. Freedays are taken into account by the contract work rate.

**Note:** The cost element for separation of standby should preferably be used instead of this fairness element. No history is taken into account and this makes fairness where the targets are very small unreliable.

### Block time

**Element:** BLOCK TIME

**Subelement:** N/A

The fairness element tries to distribute the number of block time minutes for crew.

The work rate for BLOCK TIME is reduced by non-freeday activities that don’t contribute to the BLOCK TIME element. Pre-assigned freedays are transparent to the work rate calculation. Freedays are taken into account by the contract work rate.

#### Layovers

**Element:** LAYOVER

**Subelement:** Airport name (e.g. EWR)

The fairness element tries to distribute the number of layover at specified airport for crew. This is primarily aimed at distributing destinations for LH crew and to be used in connection with long term fairness.

### Cabin crew specific elements

These elements are special fairness elements implemented for cabin crew to accommodate fairness that takes into account that long haul and short haul production shall be distributed and that there are crew that do not fly long haul production. It also takes into account that short haul work shall be shared between the AS and AH ranks.

AP rank is treated as a special rank while AS and AH targets are connected to each other. For the AP rank a fulltime target for long haul crew to only fly long haul production and one fulltime target for short haul crew only flying short haul production is calculated. Obviously the short haul crew gets a much higher target. It is decided to even out the difference by assigning a short haul target to the long haul crew so that the total work target is the same for the two groups.

Similar care is taken in respect to the AS and AH rank. In this combined case a long haul target is calculated for AS and AH crew with long haul qualifications. This is possible since AS is not supposed to fly long haul production as AH. Then short haul targets are calculated and distributed so that the total work target for all AS and AH is the same.

#### Cabin crew duty days

**Element:** CC DUTY DAYS

**Subelement:** SH, LH and TOT

The fairness element tries to distribute the number of on-duty calendar days for crew.

The work rates for CC DUTY DAYS are reduced by off-duty activities that are not considered freedays. Pre-assigned freedays are transparent to the work rate calculation. Freedays are taken into account by the contract work rate. All other activities (on-duty) shall contribute towards the DUTY DAYS goal.

The CC DUTY DAYS element requires the fairness crew group setting to contain only rank and long haul belonging, i.e. “AP,KL”, “AP,KK”, “AS,KL”, ... .

#### Cabin crew duty time

**Element:** CC DUTY TIME

**Subelement:** SH, LH and TOT

The fairness element tries to distribute the number of on-duty minutes for crew.

The work rate for CC DUTY TIME is reduced by off-duty activities that are not considered freedays. Pre-assigned freedays are transparent to the work rate calculation. Freedays are taken into account by the contract work rate. All other activities (on-duty) shall contribute towards the CC DUTY TIME goal.

The CC DUTY TIME element requires the fairness crew group setting to contain only rank and long haul belonging, i.e. “AP,KL”, “AP,KK”, “AS,KL”, ... .

### Flight Crew Specific Elements

#### Two pilot trips

**Element:** TWO PILOTS

**Subelement:** N/A

The fairness element tries to distribute the number of two pilot flights for crew.

This only applies to LH pilots and the aim is to use it in connection with long time fairness.

## Reports

The following fairness reports are available and documented in the ‘Functional Reference –Reports’:

* Fairness Info – A list of each crew’s work rate, target and achieved values.
* Fairness Statistics – An aggregated statistics view on targets and deviations for all crew in the window.

# Preferential Bidding System – PBS

This chapter describes the PBS installation at SAS and its components and the planning tools that support the process.

## Introduction

PBS (Preferential Bidding System) allow crew to influence their rosters; what kind of pairings they are assigned, and which days off they are given. CCR will automatically attempt to fulfil the bids, considering fair distribution. There are many ways of implementing PBS; the one implemented in this installation is the Lifestyles system.

The lifestyle system is the new way for crew to describe their preferences in a general way, while retaining the possibility to express specific preferences like days off or desired pairings.

This system is based on the concept of fulfillment, measured by a percentage between 0% and 100%. The fulfillment level describes how many days of that roster align with crew’s chosen lifestyle (see section 5.2.1 for details). A crew member with 30% fulfillment is less satisfied than a crew member with 80% fulfillment.

The system aims to achieve 100% fulfillment for every crew. Due to quadratic penalties for being below the target (100% for every crew), it will try to “balance out” the fulfillment among all crew. Those penalties are a source of fairness across the crew group.

## Lifestyles

Lifestyle selection is a central concept in a Lifestyles bidding system. Each lifestyle poses a set of requirements on the shape of the roster and / or the kind of pairings that should be assigned to it.

### Lifestyle Fulfillment

Lifestyle fulfillment measures how the workload on the roster fulfills the lifestyle bid for crew. It is a ratio of days on crew’s roster that fulfill the lifestyle to all working days in period with exception of days overlapping preassigned activities. Note that fulfillment includes all non-preassigned trips, even those that simultaneously fulfill both a dated bid.

### Lifestyle choices

The crew can select one of the fourteen available lifestyles. The following list contains a short description and relevant parameters for each of them:

#### Morning Person

Morning Person prefers duties that start before a certain hour, latest possible starting hour is a parameter.

Parameter: Morning Person: latest trip start

Latest allowed trips start for trips fulfilling a Morning Person lifestyle.

#### Evening Person

Evening Person prefers duties that start after a certain hour. The earliest starting hour is parametrized.

Parameter: Evening Person: earliest trip start

Earliest allowed trips start for trips fulfilling an Evening Person lifestyle

#### Nights at Home

Crew with Nights at Home lifestyle prefer short trips with night stops at homebase and few nights at outstations.

#### Nights at Home + Morning person

Crew with Nights at Home and Morning Person lifestyle prefer short trips with night stops at homebase and duties that start before a certain hour, latest possible starting hour is a parameter.

#### Nights at Home + Evening person

Crew with Nights at Home and Evening Person lifestyle prefer short trips with night stops at homebase and duties that start after a certain hour, earliest possible starting hour is a parameter.

#### Nights Away from Home 2-3 days

Crew with Nights Away from Home 2-3 days lifestyle prefer trips that are 2-3 days longs and have few nights at homebase.

#### Nights Away from Home 2-3 days + Morning Person

Crew with Nights Away from Home 2-3 days and Morning Person lifestyle prefer trips that are 2-3 days longs, have few nights at homebase and duties that start before a certain hour, latest possible starting hour is a parameter.

#### Nights Away from Home 2-3 days + Evening Person

Crew with Nights Away from Home 2-3 days and Evening Person lifestyle prefer trips that are 2-3 days longs, have few nights at homebase and duties that start after a certain hour, earliest possible starting hour is a parameter.

#### Nights Away from Home 3-5 days

Crew with Nights Away from Home 3-5 days lifestyle prefer trips that are 3-5 days long and have few nights at homebase.

#### Nights Away from Home 3-5 days + Morning Person

Crew with Nights Away from Home 3-5 days and Morning Person lifestyle prefer trips that are 3-5 days longs, have few nights at homebase and duties that start before a certain hour, latest possible starting hour is a parameter.

#### Nights Away from Home 3-5 days + Evening Person

Crew with Nights Away from Home 3-5 days and Evening Person lifestyle prefer trips that are 3-5 days longs, have few nights at homebase and duties that start after a certain hour, earliest possible starting hour is a parameter.

#### Longhaul West/U.S

This lifestyle is matched by trips that touch airports which are located geographically west of the home base. Note: “east” and “west” definitions are not parametrized. Instead, more than 4 hours difference in time zones is required.

#### Longhaul East/Asia

This lifestyle is matched by trips that touch airports which are located geographically east of the home base. Note: “east” and “west” definitions are not parametrized. Instead, more than 4 hours difference in time zones is required.

#### Longhaul Any

This lifestyle is matched by trips that touch airports which are located geographically east or west of the home base. Note: “east” and “west” definitions are not parametrized. Instead, more than 4 hours difference in time zones is required.

## Bids

In addition to specifying a lifestyle, crew can place up to three dated bids to further customize their roster preferences. The available bids act as a complement to lifestyle choices and therefore contain no option of specifying max occurrence( "I would like this bid to be fulfilled max 3 times") nor re-occurrence pattern (“I want to be off on Fridays”).

Each of the three bids can be placed on one of three priority levels: LOW, MEDIUM or HIGH (up to 1 bid per priority level).

Some of the bids are only available for specific crew groups, for example variable and fixed group. Such restrictions are handled by InterBids and are not handled by the CCR system nor covered in this documentation.

The following bids are available in the system:

### Time off

Crew bid for a personal time off. This bid is granted if there are no on-duty activities during the specified period of time.

### Stop

Crew bid for a stop at a particular station. The stop has to have of at least minimum length, controlled by a parameter. Additionally, this bid is granted on leg level, so a trip can grant the bid more than once.

Parameter: Min connection time to consider stop for stop bid

This parameter controls the minimum connection time to consider the appropriate stop bid grated. Default value: 10:00.

### Flight

Crew bid for a specific flight (identified by carrier + flight code) on a given day.

### Compensation day bids

Even though the bids for compensation days are part of the same bid file as the regular bids they are not handled by the PBS functionality, they are pre-assigned beforehand. See chapter 7 on Compensation days for more details.

## Cost function

On a high level, lifestyle fulfillment and bid achievement are modelled as two separate costs. Their importance for the optimizer can be adjusted using the weights:

Parameter: Weight of LIFESTYLE

Parameter: Weight of BIDS

Both the lifestyle fulfillment cost and the bid achievement cost work in a similar way in the sense that they both define a target and a cost that is a function of the distance of how far a crew member’s fulfillment is from the target. The cost function is the same for both cost elements and, for each of the cost elements, it internally consists of two factors, one that depends linearly on the distance from the target, and one that depends quadratically on the distance from the target.

The linear part of the cost function is a penalty when the crew member’s fulfillment is below target, it is 0 when the crew member’s is on the target and is negative (i.e., it is a reward) when the crew member is above the target. This means that the optimizer will always strive to increase the crew member’s fulfillment, even if the crew member is already above the target.

The quadratic part of the cost function works only when the crew member’s fulfillment is below its target and is a penalty that is the bigger the further the crew member is away from their target. The fact that this penalty is quadratic means that it will be better for the optimizer to leave two crew members a little below target than leaving one of them on or above the target and letting the other one be much below target, which means that the optimizer strives for a fair distribution of fulfillment. As an example, if the target for lifestyle fulfillment is 100 and the optimizer has a choice between a solution where two rosters would be 95 fulfilled and a solution where one roster would be 99 fulfilled and the other would be 91 fulfilled, for the 95 fulfillment solution, the penalty for deviating from the target would be (100 (the target) - 95 (crew fulfillment))^2 \* 2 (two crew members) = 50, while the penalty for the 99/91 fulfillment solution would be (100-99)^2 + (100-91)^2 = 1 + 81 = 82. The optimizer would prefer the 95 fulfillment solution because of its lower cost.

Both factors of the cost function can be weighted using the following parameters:

Parameter: Linear coefficient for reward per point above target

Parameter: Quadratic coefficient for cost per bid point below target

In the cost function cost(deviation) = a \* deviation^2 + b \* deviation, the above parameters are a and b.

## System configuration

### Period

During the planning process the planning period end parameter is often set to the 10th or similar in next month to be able to better plan the end of the month. Bids however should not be granted during the next month because there is a high chance of them not being granted when performing the planning for the next period.

For bids the publication period end is used, which is always one calendar month after the start of the planning period. A bid is relevant for the planning period if the bid validity period overlap the publication month.

For roster bids, time off etc., there is the possibility of using a short carry out window after the publication period because otherwise it can be difficult to grant them when they overlap the end of the month. It is optional, and default is 48:00.

Parameter: Publication period carry out for roster bids  
This is used for interpreting roster bids, they are valid this amount of time after the publication period

### Further system configuration

Further system configuration is done with code-changes in rave and python. This process is described in the System Reference.

## The PBS process

The process of including the crewmembers bids in their rosters includes the following major steps:

1. The crewmembers use InterBids to enter their bids.
2. InterBids gather the bids of all crewmembers and deliver them to the CCR system.
3. The system takes into account the bids crew placed when assigning pairings / activities.

These steps should be performed from Studio side in order to use PBS in the rostering process. Each step is listed together with the relevant parameters and commands.

* 1. Select the appropriate bid table coming from InterBids

Normally the planner would not need to change or select the bid table coming from InterBids. For testing and simulation purposes however it might be useful to be able to select a different bid table.

Relevant parameters, commands and tables:

Parameter: ETAB: Bid table  
This should point to the appropriate bid table. Use the command below to change it.

Command: Planning Tools -> Bid table selection… -> Select bid file…

Use this command to select which bid file to use. The default “bids” is the most up-to-date bids for the current rostering period. This is updated from the 16:th to the 6:th every month.

* 1. Setup the roster bid carry out window

Normally, the planner would not change this parameter. If the bid behavior around publication period end is satisfactory then this parameter is best left at its default value. If too many roster bids are perceived to be granted when planning month 1 but then not granted after planning month 2 then this parameter could be set to a smaller value to minimize this effect.  
  
Relevant parameters, commands and tables:  
  
Parameter: Publication period carry out for roster bids  
This is used for interpreting roster bids, they are valid this amount of time after the publication period

* 1. Perform the production run:
     1. Select the input crew and trips
     2. Adjust cost function parameters for Unassigned, Fairness, Quality, Stability, Lifestyles and Dated Bids.
     3. Start an optimization job and wait for it to finish.
  2. Post optimization analysis using reports, filters and sorts

After the optimization job has finished there are a number of reports, filters and sorts available to help analyze the result. Report, filters and sorts are available from the normal menus

## Planning tools

There are various tools that supports the PBS process such as reports, filters and sorting.

### Reports

The following PBS reports are available and documented in the ‘Functional Reference – Reports’.

* Lifestyle / Bid Statistics – *LSBidStatistics.py* – Displays general bidding statistics, lifestyle distribution and statistics on each bid type
* Crew Bid Outcome – *CrewBidOutcome.py* – Displays lifestyle and bidding statistics for crew, along with some general bidding statistics. Intended for the crew members. Note: the report uses fulfilment metric.

### Filters

There are filters for finding crew and trips.

#### Lifestyle Filters

The following lifestyle filters are available:

* Have Lifestyle Bid – filter all crew with a lifestyle bid in period.
* Lifestyles… - Filter crew with specific lifestyle.
  + Morning person
  + Evening person
  + Nights at home
  + Nights at home + Morning person
  + Nights at home + Evening person
  + Nights away from home 2-3 days
  + Nights away from home 2-3 days + Morning person
  + Nights away from home 2-3 days + Evening person
  + Nights away from home 3-5 days
  + Nights away from home 3-5 days + Morning person
  + Nights away from home 3-5 days + Evening person
  + Longhaul West/U.S
  + Longhaul East/Asia
  + Longhaul Any

#### Bid Filters

There are three bid-related filters:

* Bids in pp – This is a fixed filter that filters all crew with valid bids in period.
* Roster point ratio:
* Above target – This is a fixed filter that filters all crew that have a bid ratio above their target
* Below target – This is a fixed filter that filters all crew that have a bid ratio below their target

### Sorts

Crew can be sorted by Lifestyle Fulfillment or Bid Points. The sorts are accessible from the “Assignment General” menu.

# Optimizer Cost function

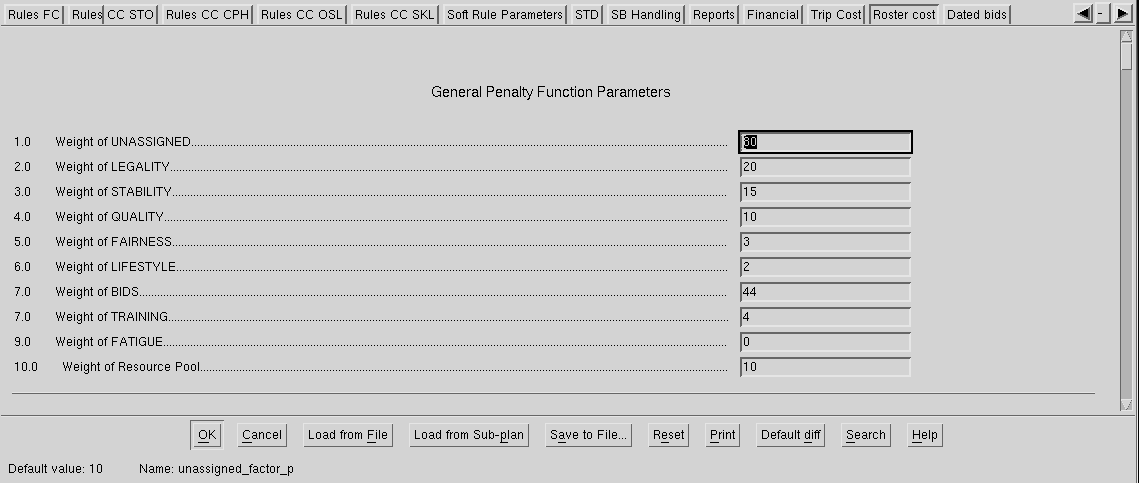
The optimizer cost function on the rostering level only deals with soft costs, i.e. costs that will not be paid directly (compared to real costs that will have to be paid directly). They are nevertheless important since they, in many cases, will incur real costs at a later stage or at a different level. For example, quality costs such as a penalty for late check-outs will not have any real costs associated with them at the present stage. However, if we don’t control them it might lead to exhaustion of crew and costly sick-leaves. The terms “penalty” and “soft cost” are mostly interchangeable and all cost elements in this section can be thought of as penalties.

## Planning period

The costs and penalties are only calculated for activities touching the planning period, any assignment after the end of the planning period will be “free” as far as the rostering optimizer is concerned. This implies that any trips used in the optimizer input should touch the planning period.

## Parameters

The parameters for the various roster cost elements are collected in the parameter form under the tab “Roster cost”. The names displayed in this document are the same that are shown in the parameter form shown in Figure [10].





## Cost elements

The cost elements in this section are what guide the optimizer in creating the best possible rosters. Parameters can be used to influence the solutions given (i.e. what is “best”). They are partitioned in a few sections.

### Unassigned

Leaving a trip unassigned should incur costs for the optimizer, since this is the force that makes it assign trips to rosters. In some instances it might be hard to assign everything, and in that case we need some granularity to enable the optimizer to decide which trips are best left open. The costs vary with the number of crew positions left unassigned.

Parameter: 1.0 Weight of UNASSIGNED

The influence of unassigned costs on the total roster cost function is controlled by this weight. When all costs associated with unassigned trips are calculated they are multiplied by this weight before being added to the total cost of the plan.

Parameter: 1.0.1 Factor for reduced cost of MANKO trips

Trips marked with *MANKO* are considered to be better than others and should incur a lower cost if left unassigned since they'll be easier to assign afterwards (i.e. to standby crew). The cost is calculated as usual and then divided by this parameter.

Parameter: 1.0.2 Factor for reduced cost of trips outside of published pp

The optimizer uses a larger time span than the published planning period, but trips starting outside of the published period should incur a lower cost since they could possibly be handled in the next period. The cost is calculated as usual and then divided by this parameter.

#### Unassigned long flight duty trips

We separate long trips from short since they typically are harder to assign to crew afterwards. The cost is controlled by three parameters.

Parameter: 1.1.1 Limit (days) for long flight duty trips

Trips of this at least this length are considered to be long in the related calculations.

Parameter: 1.1.2 Long flight duty, daily penalty

The penalty (cost) for each day of a long, unassigned trip.

Parameter: 1.1.3 Long Flight duty (quadratic on days), daily penalty

This penalty is multiplied by the number of days, squared, of the trip, generating behaviour where leaving two 3-day trips open is less costly than leaving one 6-day trip open.

#### Unassigned short flight duty trips

This is the default case for trips, the long ones having already been sorted out. There is only one parameter since the divider between long and short trips have already been defined.

Parameter: 1.2 Short flight duty, daily penalty

Trips shorter than the limit set by the parameter Limit (days) for long flight duty trips incur this penalty for each day of the trip.

#### Unassigned charter trips

Charter trips are hard to assign afterwards and therefore incur an extra cost if left unassigned.

Parameter: 1.3 Charter, daily penalty

Trips defined (by flight number) to be charter trips are penalized by this cost for each day.

#### Unassigned troublesome trips

Some trips are defined to be “troublesome” and therefore incur a higher (or at least different) cost since they typically are harder to assign afterwards. The definition depends on ranks, AC qualifications and destination airport (and indirectly on the base for which planning is performed).

Parameter: 1.4.1 Troublesome trip, daily penalty

Trips determined to be troublesome incur this penalty for each day.

Parameter: 1.4.2 Troublesome AC qualifications

AC qualifications that are hard to assign, for instance due to limited number of qualified pilots (can be a comma-separated list). This parameter is ATM. located in tab “ALL PARAMETERS”.

Parameter: 1.4.3 Troublesome airports

Destination airports that are hard, or otherwise considered important, to assign (can be a comma-separated list). This parameter is ATM. located in tab “ALL PARAMETERS”.

#### Unassigned standby

Unassigned standby have a separate cost since they should be penalized less than regular flight duty trips. Only standby blocks are targeted by this cost, not lines.

Parameter: 1.5 Standby, daily penalty

The cost of each day of standby trips left open (unassigned).

#### Unassigned blank days

Unassigned blank days have a separate cost since they should be penalized less than regular flight duty trips.

Parameter: 1.6 Penalty for blank days

The cost of each unassigned blank day.

#### Unassigned heavy duty

Trips with heavy duties are difficult to assign, and the more heavy duties in the trip, the more difficult it is. The cost element targets this by multiplying the number of heavy duties in a trip by a penalty. The definition of a heavy duty for the purpose of this penalty is defined by a parameter.

Parameter: 1.7 Heavy duty penalty, per duty

The penalty per heavy duty in an unassigned trip

Parameter: 1.7.1 Definition of heavy duty

The amount of duty time in a duty for it to be considered heavy (for the purpose of this penalty)

#### Unassigned long haul trip

Long haul trips are prioritized and so there is an extra cost per unassigned long haul trip.

Parameter: 1.8 Penalty for unassigned long haul trip, per trip

The penalty per unassigned long haul trip

#### Unassigned training

Training are important to assign and there fixed cost per unassigned training trip that can be used to force the optimizer to assign more training.

Parameter: 1.9 Penalty for unassigned training

The penalty per unassigned training trip

### Legality costs

Legality costs are costs that are used instead of rules that can't be modelled as hard rules due to illegal sub chains during shift-improve.

Parameter: 2.0 Weight of LEGALITY

The influence of legality costs on the total roster cost function is controlled by this weight. When all costs associated with illegal trips and rosters are calculated they are multiplied by this weight before being added to the total cost of the plan.

#### Recency

Government rules specify how often a crew must perform a certain action to keep his or her recency status. When a crew becomes unrecent a penalty (cost) is incurred. This cost should be high enough to simulate hard rule behaviour, but still limited so that the optimizer can perform shift-improve actions.

Parameter: 2.1 Penalty for unrecent crew

The cost incurred for each trip assigned to crew with lost recency.

#### Mixed standby/production

There are rules limiting how standby and flight production can be combined. Due to technical issues with the implementation these rules can’t be active in the optimizer. To help the optimizer create legal solutions these rules are implemented as high costs instead.

Parameter: 2.2 Penalty for mixed SBY/PROD or SBY/BL

The cost incurred for each working period in the planning period which has standby mixed with other production.

### Stability costs

To increase the stability of generated rosters we have cost elements whose purpose is to keep the optimizer on a “safe distance” from legal boundaries.

Parameter: 3.0 Weight of STABILITY

The influence of stability costs on the total roster cost function is controlled by this weight. When all costs associated with stability elements are calculated they are multiplied by this weight before being added to the total cost of the plan.

Parameter: 3.0.0 Reduction for BL costs

Blank day costs, 3.10, 3.11, 3.12, 3.15, can be reduced with a factor.

#### Used trip rest buffer

As a safeguard for delays etc. we have a soft buffer for rest after a trip. A penalty (cost) is incurred for each minute of the buffer that is used, when also the next wop is a flight duty.

Parameter: 3.1.1 Soft buffer for legal rest after trip

The buffer size in minutes.

Parameter: 3.1.2 Penalty for each hour of used soft rest buffer

The cost of each hour of used buffer time.

#### Legs close to recency expiration date

Having crew become unrecent generates very high costs and should be avoided. Due to unforeseen events such as sickness we want the optimizer to generate rosters that are resilient to this. To be able to handle crew with more than one qualification correctly we need to influence the optimizer to distribute recency generating legs in a suitable way. This cost will punish all legs not extending the recency deadline.

**Parameter: 3.2.1 Penalty-parameter for legs close to recency deadline**

The cost for each day of used recency deadline buffer.

**Parameter: 3.2.2 Parameter for recency distance function**

The size, in days, of the recency buffer.

#### Distance to first recency expiration date

For crew with multiple qualifications it is necessary to keep all expiration dates as far into the future as possible. This cost rewards the number of days after the end of the publish period until the first expiration date of all qualifications. The volume in the report Rostering Statistics refer to the number of days. The expiration date is based on all assignments before the end of the publish period.

**Parameter: 3.3 Reward for every 10 days of recency beyond the planning period**

The reward for every 10 days after the end of the planning period that crew is recent.

#### Used 270-points buffer

As with the other soft buffers, this is used to increase resiliency against unforeseen events that would otherwise render a roster illegal. For each duty in a roster the optimizer checks the current points and calculates the cost accordingly.

Parameter: 3.4.1 Soft buffer for 270 duty points

The buffer size in points.

Parameter: 3.4.2 Penalty for each point of used soft 270 points buffer

The penalty of each point of used soft buffer.

#### Used duty-time in 7x24 buffer for SBY and BL

As with the trip rest buffer this is a buffer to keep the rosters legal even when faced with minor disturbances. It is used to keep crew at a safe distance from the maximum number of duty hours in 7 seven days (typically 45 hours).

In order to achieve rosters with more effective standby and blank day assignments, extra duty time is added to standby blocks and blank days. If the duty time, including the extra time, in 7x24 hours exceeds the limit specified by the parameter a penalty is incurred

Parameter: 3.5.1 Penalty for each hour above soft duty time limit

The cost of each used duty time hour above the soft limit.

Parameter: 3.5.2 Soft limit for duty time in 7\*24 hours with sby & bl

The soft 7\*24 duty time limit, in hours.

Parameter: 3.5.3 Additional duty time for a BL-day

This time is added to the regular duty time for a BL-day.

Parameter: 3.5.4 Additional duty time for a SBY-day

This time is added to the regular duty time for a standby day (standby blocks).

#### Freeday buffer before standby

For a standby block to be effective crew must be rested before and be able to rest after. Therefore it is desirable to have more than the 2 stipulated days off before and after a working period with much standby. This cost puts pressure on increasing the number of days off to at least 3.

Parameter: 3.6.1 Penalty for less than 3 freedays before standby

The penalty incurred if crew have less than 3 days off before a wop with standby

Parameter: 3.6.2 Threshold for standby freeday buffer cost

The number of standby days in a working period must be at least this many for the freeday buffer cost to apply

#### Freeday buffer after standby

A companion cost to Freeday buffer before standby. It uses parameter 3.6.2.

Parameter: 3.7 Penalty for less than 3 freedays after standby

The penalty incurred if crew have less than 3 days off after a wop with standby

#### Standby before free weekend

If a standby is placed directly before a free weekend, i.e. the standby ends on a Friday and no other duty is scheduled over the weekend, the last day of the standby will difficult to use efficiently since crew should not loose their free weekends.

Parameter: 3.8 Penalty for standby before a free weekend

The penalty incurred if crew have a standby day assigned on a Friday before a free weekend

#### Blank day after freedays

For cabin crew blank days should not be placed directly after a freeday period. This is because crew have no obligation to report for duty and it might be difficult to get hold of crew and thus difficult to use the blank days to cover production. This not an issue for flight deck since crew themselves must report before duty and hence the penalty should be set to 0 when planning flight deck.

Parameter: 3.9 Penalty for blank day directly after freedays

The penalty incurred if crew have blank days assigned first in a working period. The penalty is for each blank day, i.e. if crew have two blank days first in a working period the penalty is two times the value of this parameter.

#### Blank day after late check out

Blank days placed after a late check out are difficult to use, since crew cannot work until late that day.

Parameter: 3.10.1 Penalty for blank day after late check-out

The penalty for assigning a blank day after a late check out.

Parameter: 3.10.2 Max time considered as late check-out before blank day

Duties ending before this time are considered as late check-out.

#### Blank day before early check in

Blank days placed before an early check in are difficult to use, since crew cannot start working until late that day, because of the late check out the day before

Parameter: 3.11 Penalty for blank day before early check-in

The penalty for assigning a blank day before an early check in.

#### Blank day before free weekend

If a blank day is placed directly before a free weekend, i.e. on a Friday and no other duty is scheduled over the weekend, the blank day will difficult to use efficiently since crew should not loose their free weekends.

Parameter: 3.12 Penalty for blank day before a free weekend

The penalty incurred if crew have a blank day assigned on a Friday before a free weekend

#### Blank day after standby

Common for blank days and standbys are that a longer period is more effective, it can be used to cover longer trips or more trips. Thus when assigning a blank day, if possible, it is good if it can be placed adjacent to a standby or another blank day (see [6.3.3.14]), thereby creating a more flexible roster.

However, it is more difficult to efficiently use a blank day followed by a standby than the other way round. When assigning a blank day a small reward is therefore awarded if it is possible to assign it directly after a standby.

Parameter: 3.13 Reward for blank day after standby

Reward for assigning a blank day directly after a standby.

#### Adjacent blank days

A single blank day is less effective than several adjacent blank days, or standby followed by blank days (see [6.3.3.13]). Thus when assigning a single blank day, if possible, it is good if it can be placed adjacent to another blank day.

Parameter: 3.14 Reward for assigning blank days together

Reward for assigning a single blank day adjacent to another blank day.

#### Fair distribution of blank days

The distribution of blank days between crew should be fair. This cost element is used to prevent any one crew from getting too many blank days in the planning period. There is only an upper limit, there is no pressure to assign a certain minimum amount of blank days. The cost is quadratic in the number of assigned blank days above the chosen threshold.

Parameter: 3.15.1 Penalty (quadratic) for assigned blank days

This parameter sets the penalty incurred for the squared number of assigned blank days above the threshold.

Parameter: 3.15.2 Threshold for blank day assignment cost

If the number of assigned blank days is above this value there is a penalty.

#### Consecutive freedays, 50% crew

In order to get an even distribution of production (per crew) during the planning period, there is a penalty if a minimum number of freedays in a row is not achieved. A minimum value can be specified for different work rate categories. Only crew in variable group are valid for this penalty.

Parameter: 3.16.1 Penalty per freeday below consecutive target

This parameter sets the penalty incurred per day less than the target.

Parameter: 3.16.2 Consecutive freedays target for 50-74% crew

The consecutive freedays target for 50-74% crew.

#### Consecutive freedays, 75% crew

The same cost as 6.3.3.16, but for 75-99% crew. It uses the parameter 3.16.1.

Parameter: 3.17 Consecutive freedays target for 75-99% crew

The consecutive freedays target for 75-99% crew.

#### Consecutive freedays, 100% crew

The same cost as 6.3.3.16, but for 100% crew. It uses the parameter 3.16.1.

Parameter: 3.18 Consecutive freedays target for 100% crew

The consecutive freedays target for 100% crew.

#### Freedays before summer vacation

Preferably place the 4 freedays off in regard to summer vacation before and not after to increase future stability. Penalize each freeday less than 4 before a summer vacation

Parameter: 3.19 Penalty per freeday less than 4 before summer VA

This parameter sets the penalty incurred per day less than 4 before a summer vacation.

#### Freeday target

Preferably crew should have an equal amount of freedays from month to month. As an example, 100% crew must have at least 12 freedays in one month and 26 freedays in two months. The preferable freeday pattern would then be 13-13 and not 12-14.

Legality ensures that crew get at least the number of freedays that he or she should have. This penalty aims at guiding crew towards a given freeday target in one month, which might differ from the legal need. If the desired normal pattern for 100% crew is 13-13, the parameter below should be set to 13.

If in one month, for some reason, there is more or less room in the schedule, a deviation from the normal target can be specified using a parameter. This deviation is summed with the normal target to get the 100% target for a month. The targets for other crew are automatically calculated using their part time grades. The freeday target is also reduced in the same way as when calculating the normal freeday need for crew, for example when there is preassigned illness or vacation on the roster. Part time changes are not considered by this penalty, so if crew change part time grade during the month the cost will be slightly off.

The penalty is based on the squared deviation from the target. Thus, both crew with more and less than the target are punished. Also, it is more costly to let one crew with the same part time grade get 1 extra freeday and another crew get 3 extra freedays, then to let both crew get 2 extra freedays.

Parameter: 3.20.1 Penalty per freeday deviation from target (squared)

This is the penalty per squared day of deviation from the target.

Parameter: 3.20.2 Normal freeday target for 100% crew

This parameter sets the desired normal amount of freedays in one month for 100% VG crew.

Parameter: 3.20.3 Preferred deviation from normal freeday target for 100% crew

This parameter is used to specify a deviation from the normal target if the problem is more or less tight than normal. Can be both positive and negative.

#### Part time summer vacation length

For part time variable group crew we want to ensure that they get at least 22 days of summer vacations. This cost is applied for each day lacking in the vacation. For part time crew vacation consists of a combination of vacation days and freedays.

Parameter: 3.21 Penalty for each day part time summer vacation is shorter tha required

This is a linear penalty applied for each day the part time vacation is shorter than 22 days.

#### Troublesome ranks

In some cases the optimizer might find solutions that are better on an individual basis, but that fails on requirements on assigned crew with specific rank. To control this there is a reward that will help the optimizer assign troublesome ranks before other positions.

Parameter: 3.22.1 Troublesome ranks that should be preferentially assigned

The ranks that should give a reward when assigned.

Parameter: 3.22.2 Reward for each day of assigned troublesome rank trip

The reward per day.

### Quality costs

To create rosters that are more enjoyable for the crew we have cost elements that can be used to guide the optimizer towards solutions that are better from the point of view of the crew. This is not without drawbacks since the bidding process (PBS) is aimed towards giving every crew what they individually want, and this might differ from general opinions on what is desirable.

Parameter: 4.0 Weight of QUALITY

The influence of quality costs on the total roster cost function is controlled by this weight. When all costs associated with quality elements are calculated they are multiplied by this weight before being added to the total cost of the plan.

#### Late finishes before days off

Crew typically dislikes finishing late before days off since it could interfere with their plans. This cost element is used to reduce this behaviour and is simply a parameter-specified cost multiplied by the number of instances of late finishes.

Parameter: 4.1.1 Factor for increased check-in/out cost when rest is short (2 days)

If the rest after the late check-out is short, two days, then the penalty is multiplied by this factor.

Parameter: 4.1.2 Penalty for a late finish before day off

The unit penalty of a late finish.

Parameter: 4.1.3 Limit for a late finish before day off

Trips that finish after this time are considered to be late finishes.

#### Early starts after days off

This is analogous to the case with late finishes. It uses the parameter 4.1.1.

Parameter: 4.2.1 Penalty for an early start after day off

The unit penalty of an early start.

Parameter: 4.2.2 Limit for an early start after day off

Trips that start before this time are considered to be early starts.

#### Fly-below-rank, trip days

Although flying below rank increases the solution space and could be beneficial it should be limited since it should not be considered as optimal solutions. Flying below rank is only defined at trip level, but the penalty counts the number of days squared in fly-below-rank trips times the unit cost.

Parameter: 4.3 Penalty for fly-below-rank (per day, squared)

The penalty of one squared fly-below-rank day

#### Fly-below-rank separation

This penalty also applies to fly-below-rank, but instead of limiting the number of fly-below-rank days directly, it tries to increase the separation between individual fly-below-rank trips. The two costs can be used together or separately depending on planning area.

The penalty is zero when there is no previous fly-below-rank trip within a certain time frame, or as much environment as is currently loaded, and increases linearly per day the closer two fly-below-rank trips get to each other.

Parameter: 4.4.1 Penalty for fly-below-rank (per day of decreased separation)

The penalty for one day of separation less than the specified days.

Parameter: 4.4.2 Max fly-below-rank separation days

The number of days to punish fly below rank.

#### Early check-in

Less undesirable than early check-ins after days off are general early check-ins, but they still can be punished separately. In this case the total amount of time before early check-ins are summed on roster-level and then squared so that the optimizer will avoid giving individuals disproportionate amounts of early check-ins. An early check-in is one defined as taking place before 7:00 (LT).

Parameter: 4.5 Penalty for early check-in (per hour, squared)

The penalty per squared hour of time before the early check-in limit.

#### Base variants

As with fly-below-rank, base variants can be used to increase the solution space, but should in general be avoided. It’s implemented as a check whether both the first and the last leg in a trip are deadheads, the total cost is then the unit cost times the number of trips with this characteristic.

Parameter: 4.6 Penalty for base variants

The unit cost of a trip that is defined to be using base variants.

#### Standby block separation

To achieve a certain degree of fairness in standby block assignment there is a penalty that increases with decreasing distance to the previous standby block, rounded down to whole weeks. A parameter decides how many weeks separation that is necessary for the penalty to apply, beyond this limit there is no penalty for assigning a standby block.

There are two components of the cost function:

* A linear component that starts with a step at the soft limit. The step is intended to make it much more awarding to assign a standby block just beyond the limit instead of just before. This component increases linearly with decreasing distance to the previous standby block.
* A small quadratic component that increases quadraticlly with is intended to perform a fairness pressure. It works by making it cheaper to assign a standby block to two crew in the middle of their penalty range, instead of assigning a standby block close to the previous for one crew and far away (but still within the limit) for the other one.

Tuning this cost requires three parameters: the limit L, the linear component M and the quadratic component Q.

* Tune L towards the desired separation.
* Tune M to increase/decrease the pressure towards the separation target L
* Tune Q to increase/decrease the fairness pressure. Q yields a smaller absolute cost than M per unit, and is meant only as a fairness component, not for replacing M as separation pressure

The mathematical formula:

Penalty = D \* M + D \* sqrt(D) / 10 \* Q

where D = max(L-W, 0)

W = number of weeks from previous standby block

Parameter: 4.7.1 Soft limit for standby block separation (weeks)

The preferred number of weeks (referred to as L above) between standby blocks. Beyond this limit the cost will be 0

Parameter: 4.7.2 Standby block separation: separation pressure

Linear cost (refereed to as M above) per week closer to the previous block than the limit. The linear cost includes a step that is 10\*M. This means that the linear cost is 0 at week W but 10\*M at week W-1

Parameter: 4.7.3 Standby block separation: fairness pressure

Quadratic component (referred to as Q above) used to achieve fairness in standby block assignment in the range 0-W weeks.

#### Standby line separation

Standby line separation works the same way as standby block separation, see [6.3.4.7] for more details.

Parameter: 4.8.1 Soft limit for standby line separation (weeks)

This is the preferred number of weeks between standby lines. Beyond this limit the cost will be 0

Parameter: 4.8.2 Standby line separation: separation pressure

Linear cost per week closer to the previous line than the limit.

Parameter: 4.8.3 Standby line separation: fairness pressure

Quadratic component used to achieve fairness in standby line assignment in the range from zero weeks separation up to the soft limit.

#### Repeated layover city

It is possible to define a set of cities where it is not desirable to have layovers more than once in the planning period. Repeated layovers in any of these cities will be penalized.

Parameter: 4.9.1 Penalty for repeating layover city

The penalty per layover more than one in a city included in the set

Parameter: 4.9.2 Set of cities where more than one layover is penalized

A set of cities where multiple layovers in the planning period is penalized

#### Far west before far east

Flying a far west flight shortly before a far east flight is not illegal but the optimizer should not find it attractive to assign this pattern. It should however be possible to assign manually, hence a very high cost is needed.

Parameter: 4.10.1 Penalty for flying far west before far east within time limit

The penalty for breaking the far west-far east limit.

Parameter: 4.10.2 Soft limit (days) for far west before far east penalty

If the time between a trip to a far west destination and a trip to a far east destination is less than this limit there will be a penalty

Parameter: 4.10.3 Set of airports that constitutes far west destinations

A set of destinations considered to be located far west

Parameter: 4.10.4 Set of airports that constitutes far east destinations

A set of destinations considered to be located far east

#### Repeated long haul destination

To ensure a spread of destinations when planning long haul production there is a penalty for assigning consecutive same destinations more than two.

Parameter: 4.11 Penalty for consecutive same LH destination > 2 times

The penalty per trip in a row to the same long haul destination more than 2.

#### Duty time excess FD

Duty time in calendar month should not exceed 160 hrs per calendar month for full time crew and 9:15/9:30 for part time VG/FG respectively.

Parameter: 4.12 FD SKS: Penalty for duty time in excess of monthly or avg daily limit

The penalty per hour of excess duty time.

#### Duty time excess calendar week CC

Duty time in calendar week should not exceed 42 hrs per calendar week for cabin crew.

Parameter: 2.6 Cost for exceeding duty time per calendar week [CC]

A one-off penalty if the duty time exceeds the limit.

Parameter: 2.6.1 Max duty-time per duty in calendar week [CC]

This is the limit for exceeding duty time per calendar week. Default value 41:00.

#### Duty time excess calendar month CC

Duty time in calendar week should not exceed 185 hrs per calendar month for cabin crew.

Parameter: 2.7 Cost for exceeding total duty time per calendar month [CC]

A one-off penalty if the duty time exceeds the limit.

Parameter: 2.7.1 Max duty-time per duty in calendar month [CC]

This is the limit for exceeding duty time per calendar month. Default value 185:00.

#### 2 pilot flights

2 pilot destinations for long haul pilots must also have an element of fairness in planning. Number of 2 pilot destinations / month shall be distributed over the pilot corps.

Parameter: 4.13.1 FD SKI: Limit for number of flights with 2 pilots

The limit for when 2 pilot flights will be penalised.

Parameter: 4.13.2 FD SKI: Quadratic penalty for each flight with 2 pilots over limit

The penalty for adding more 2 pilot flights.

### Fairness costs

The fairness cost is the cost for crew to be deviating from their fairness target. The costs for individual cost elements as well as targets are set in etables. Please refer to section 2.3.3 above.

Parameter: 5.0 Weight of FAIRNESS

The influence of fairness on the total roster cost function is controlled by this weight. When all costs associated with fairness are calculated they are multiplied by this weight before being added to the total cost of the plan.

### Dated bid cost

The dated bid cost element guides the optimizer towards rosters that fulfil bids placed by the crew members. If a crew member is below their target a quadratic cost is incurred and if they are above target a linear negative cost (i.e. a reward) is incurred. See chapter 5.4 on PBS Cost function for more details.

Parameter: 7.0 Weight of BIDS

The influence of bids on the total roster cost function is controlled by this weight. When the cost associated with bids is calculated, it is multiplied by this weight before being added to the total cost of the plan.

Parameter: Factor between weights  
Defines the distance between bid weights. Taking a LOW stop bid and a MEDIUM timeoff bid as an example, this factor defines how many stops matching the stop bid it would take for the optimizer to see a roster with those stops as equivalently good as a roster only granting the timeoff bid.  
  
Parameter: Coefficient for linear penalty/reward when deviating from target  
Defines how important the linear component of the cost function should be for the bid cost.  
  
Parameter: Coefficient for quadratic penalty when below target  
Defines how important the quadratic component of the cost function should be for the bid cost.  
  
Parameter: Min connection time to consider stop for stop bid  
Defines how long a layover should be in order to be able to grant a stop bid.

### Lifestyle cost

The lifestyle cost element guides the optimizer towards rosters that fulfil lifestyles chosen by the crew members. If a crew member is below their target a quadratic cost is incurred and if they are above target a linear negative cost (i.e. a reward) is incurred. See chapter 5.4 on PBS Cost function for more details.   
  
Parameter: Commuter: Min rest period length  
Minimal length of a rest after wop in order for the rest to be seen as fulfilling the commuter lifestyle  
  
Parameter: Commuter: Min trip length  
Minimal trip length in order for the trip to be seen as fulfilling the commuter lifestyle

Parameter: Nights at home: Max nights away from base per trip  
Maximal number of nights that is allowed for a pairing to leave a crew member away from their homebase in order for this pairing to be seen as fulfilling the nights at home lifestyle  
  
Parameter: Nights at home: Max trip length  
Maximal length of a pairing in order for it to be seen as fulfilling the nights at home lifestyle

Parameter: Morning person: latest trip start  
Latest possible start of a pairing in order for it to be seen as fulfilling the morning person lifestyle  
  
Parameter: Evening person: earliest trip start  
Earliest possible start of a pairing in order for it to be seen as fulfilling the evening person lifestyle

### Training costs

Parameter: 7.0 Weight of Training

The influence of Training elements on the total roster cost function is controlled by this weight. When all costs associated with Training are calculated they are multiplied by this weight before being added to the total cost of the plan.

Parameter: 7.0.1 Training to consider in cost function   
A set with all the documents (types of REC) that shall be considered by the cost function. Can be any of OPC, PC, OPCA3, PCA3, OPCA4, PCA4, PGT, REC, RECSKN, PGTSKN.

#### Missing recurrent assignment

Parameter: 7.1 Missing recurrent assignment, penalty   
This penalty is incurred for every crew that must have any of the training activities specified in Training to consider, but hasn’t been assigned it before the expiry.

#### PC/OPC extending recency

Parameter: 7.2 PC/OPC extending recency, reward

A reward for each day a simulator extends the recency. It will only work well when the assignment will affect the recency, i.e. when the roster is mostly empty.

#### Optimal date for recurrent activity

Recurrent activities should be assigned close to the expiry, but with a safety margin, to enable a good distribution between crew. Crew should be prioritized the closer they are to expiry (i.e. crew with September as rec month should get an assignment before crew with October, of the planning month is August). This cost will give 80% of the reward in the last number of days before the expiry as defined by a parameter. 100% of the reward will be given in the expiry month before the safety margin. 60% of the reward will be given in the month before the expiry month, and 30% of the reward will be given in the month two months before expiry.

Parameter: 7.3.1 Optimal date for recurrent activity, reward

This parameter controls the safety margin.

Parameter: 7.3.2 Days before expiry considered as optimal date

The number of days before the expiry that is considered as optimal date.

#### Unnecessary ASF

Parameter: 7.4 Unnecessary ASF, penalty

ASF:s should be used for FR to retain recency. This penalty is incurred for every day below 90 that an ASF extends recency. I.e. if the simulator extends the recency with 10 days, the penalty will be 10 times the parameter value.

#### Correctly assigned AST

Parameter: 7.5 Correctly assigned AST, reward

If the crew is in need of an AST before the simulator, this element will reward the assignment of it in the correct interval (e.g. 30 days before the simulator).

#### New flight reward

A reward for every NEW flight assigned to crew that is NEW according to their training programs. The purpose of this reward is to make sure that enough NEW flights is assigned to crew. The reward is applied linearly for every flight assigned with the right qualification up to the number of required flights in the training program plus a number of buffer flights that can be different for long and short haul.

Parameter: 7.6.1 Assigned NEW flights, reward

This is a small number that is applied linearly for every suitable flight assigned to crew roster.

Parameter: 7.6.2 Short haul: Additional new flights to receive reward   
Number of flights in addition to the number of required flights to receive the reward for short haul crew.

Parameter: 7.6.3 Long haul: Additional new flights to receive reward   
Number of flights in addition to the number of required flights to receive the reward for long haul crew.

#### Missing NEW flight penalty

If the number required number of NEW flights are not assigned for the period when the NEW restriction ends a penalty is applied,

Parameter: 7.7 Missing new flights, penalty  
A large penalty that is applied when the number of assigned NEW flight is to few before end of NEW restriction/training.

#### PC/OPC assists

This cost should limit the number of assists used.

Parameter: 7.8 PC/OPC assists, penalty  
Penalty for each used assist.

## Constraints

This section covers constraints working on the whole solution.

### Standby lines – assignment target

For a base, rank, start time of day and validity period combination there can be a daily target number of assigned standby lines that should be achieved. For each combination the system created two vertical constraints, or costs, that are used to keep the assigned value close to the target. There are two constraints because it should be more expensive to assign one F.T.P/F.T.E. below the target than to have an overshoot of one F.T.P/F.T.E., thus two different costs are needed.

#### Constraint: 8.1 Standby line need upper constraint

Activate or deactivate the constraint pushing the number of assigned standby lines down towards the target

#### Parameter: 8.1.1 Cost of assigning one hundredth of a FTE above standby line target need

Cost of assigning one hundredth of a FTE above standby line target need

#### Constraint: 8.2 Standby line need lower constraint

Activate or deactivate the constraint pushing the number of assigned standby lines up towards the target

#### Parameter: 8.2.1 Cost of assigning one hundredth of a FTE below standby line target need

Cost of assigning one hundredth of a FTE below standby line target need

#### Table: standby\_line\_need

This table is used to configure the constraints (see [8.1.5.1])

### Standby lines – max single qualified crew

There is also a constraint for limiting the number of single qualified crew, in total, that can be assigned standby duty on each day in the planning period. The maximum number is adjusted via a parameter; the constraint can also be turned off altogether.

#### Constraint: 8.3 Standby lines, max single qualified crew per day

Activate or deactivate the constraint that limits the number of single qualified crew

#### Parameter: 8.3.1 Maximum number of single qualified standbys per day

The maximum allowed number of single qualified crew that are assigned standby. The limit is enforced per day in the period.

### Restricted crew on flight

First constraint ensures that matador can’t assign several restricted crew on the same flight. The limit is one crew for all flights except long-haul and BU flights were two restricted crew may be rostered on the same flight.

Second constraint ensures that only one resource pool crew can be assigned on a flight. This was made for use by CC SKS, but will work to limit temp crew from any area.

#### Constraint: 8.4 Restricted, max nr of restricted crew on flight

Activate or deactivate the constraint that limits the number of restricted crew

#### Constraint: 8.4.1 Max nr of temp crew on flight

Activate or deactivate the constraint that limits resource pool crew on flight.

### Non SCC on flight

This constraint ensures that at most one non SCC crew is assigned to a CRJ flight.

#### Constraint: 8.5 SCC, max one non SCC on flight

Activate or deactivate this constraint.

### Required crew on OPC

Some simulators require at least a fixed number of crew to be performing OPC, which is the same as a maximum allowed number of crew on PC.

#### Constraint: 8.6 Max allowed crew on PC

Activate or deactivate this constraint.

### Simulators should be fully assigned

This constraint uses a cost to limit the number of used simulators, i.e. using one 1/1/… simulator for two crew is better than to leaving one position unassigned on both simulators.

#### Constraint: 8.8 Simulator should be fully assigned

Activate or deactivate this constraint.

#### Parameter: 8.8.1 Cost of each used simulator

The cost of each simulator with any crew assigned.

### Required crew on 2x2h OPC

Simulators that don’t have a strict requirement on the numner of crew doing OPC can use this constraint to mimic the behaviour.

#### Constraint: 8.7 Soft: Max allowed crew on 2x2h PC

Activate or deactivate this constraint.

#### Parameter: 8.7.1 Cost for soft constraint that requires 1 opc on 2x2h PC

### Max crew on PGT

Using the table *pgt\_need* a certain allotment of PGT slots within a base can be defined. Apart from using this allotment in reports for follow up, a constraint can be activated for Matador. For a base, a qualification and validity period combination there can be a max number of allowed assignments during a day.

For more information, please see the training functional reference.

#### Constraint: 8.9 PGT Allotment constraint

Activate or deactivate the PGT allotment constraint pushing the number of assigned PGT slots down towards the target

#### Parameter: 8.9.1 Cost for each PGT slot assigned above maximum allowed

Cost for each PGT slot assigned above maximum allowed

#### Table: pgt\_need

This table is used to configure the constraints.

# Compensation days

The codes that are defined as compdays are specified internally in the system. They are: F0, F3, F3S, F31, F32, F35, F7S. In addition to these there are LA42 days. These will never be considered as compdays, but they can be assigned with the assign function.

## Parameters

The compdays functionality uses one parameter.

* **Compdays to assign** The codes for compdays that should be assigned by the assign compdays function

## Filters

Included in the functionality are three filters, available through **Assignment general-> Filter -> Compensation days**.

* **Compday bids** Filters crew that have compday bids
* **Unfulfilled compday bids** Filters crew that have compday bids that haven’t been granted
* **High balance** Filters crew with a compday balance higher than the limit at which point compdays can be forced. The balance for a compday type is defined as the minimum, from the start of the planning period and for all future times.
  + **Flight crew, LH: F3 + F3S + F31 > 7**
  + **Flight crew, SH: F3 + F3S > 5**
  + **Cabin crew: F0 + F3 > 7**

## Assign function

The function to assign compdays based on the crew bids is available using **Assignment general -> Create -> Compensation days (bids)…** A confirmation dialog will be shown, after which the function will try to assign any compday bid with a type in the parameter set *Compdays to assign*, for all crew shown in window. If the function encounters errors they will be written to a log which is shown when the function finishes. There are three types of errors:

* **Low balance** When there isn’t enough balance in the compday account to assign the compday bid
* **Overlapping trip** When there already exists an activity that overlaps the bid
* **Unknown reason** When the function fails for unknown reason

If the function finishes without errors a message with that content is displayed.

# Assignment functions

## Standby

The Standby function ensures that a certain amount of standby days are assigned to crew in the planning period. Standby days are either scattered, part of a standby block or part of a standby line. Most rules regarding standbys only apply to standby blocks. Standby blocks and standby lines are assigned by the optimizer, scattered standby are manually assigned post-optimization.

### Standby codes

The activity codes for standby consist of three parts:

* **A, R, RL** The first part is a letter stating which type of standby it is
  + **A:** Airport standby, block or scattered standby performed at the airport
  + **R:** General standby, block or scattered standby performed at home
  + **RL:** Standby line, long haul standby performed at home
* **2...9 [optional]** The second part is an optional qualification code, which must match crew qualification for an assignment to be legal. It can be left out and then any crew can pick up the standby. This part is not used with the standby line code (RL).
  + **2:** A320/321
  + **3:** B737-700 NG
  + **4:** A340
  + **5:** F50
  + **6:** A330
  + **7:** CRJ900
  + **8:** MD80
  + **9:** B737-4/500 (classic)
* **S [optional]** The third part states if the standby should be regarded as scattered or not. A scattered standby can assigned more freely than standby blocks/lines (most rules do not apply to scattered standby)

For example: R3S is a scattered standby at home for crew with a qualification matching B737.

### Rules

The rules handling standby are (see the Rule Overview document for details):

* Standby duration accumulated over a duty must not exceed 10 hours
* Maximum duration of standby block is X consecutive days
* Max standby days in calendar year (CABIN CREW)
* Maximum one airport standby in standby block (CABIN CREW SKD)
* Minimum 7 calendar weeks between airport standby (CABIN CREW SKD)
* Crew must have at least 3 months of experience before performing standby (CABIN CREW SKD)
* Crew must have at least 2 months of experience or 3 long-haul trips before performing standby (CABIN CREW SKD)
* Max scattered standby days in calendar year (CABIN CREW)
* Maximum length of standby lines must be 28 days (CABIN CREW SKS)
* Airport standby duration must not exceed 4 hours (CABIN CREW SKD SKN)
* No standby allowed immediately after vacation or leave of absence exceeding 7 days (CABIN CREW SKD)
* Max standby lines in calendar year (CABIN CREW SKD)
* Minimum days between standby / Minimum days between standby lines (CABIN CREW, functional)
* No illegal standby for cabin crew (CABIN CREW, functional)
* Standby must cover all days between freedays for fixed group cabin crew (CABIN CREW, functional)
* Do not mix standby duty with other duty between freedays (CABIN CREW, functional)
* No standby allowed as first duty after long period of either illness or leave of absence (CABIN CREW, functional)
* No standby for crew that has not fulfilled recency requirements (CABIN CREW, functional)
* No long-haul standby for crew that has changed from rank AH to AS within 3 months after change unless 3 long-haul flights have been completed (CABIN CREW, functional)
* No standby for crew after change to new qualification (CABIN CREW, functional)
* Maximum number of isolated standby days per calendar month (CABIN CREW SKN)
* Max standby blocks in quarter (CABIN CREW SKN)
* Airport standby following off-duty must start no earlier than 06:00 (CABIN CREW SKN)
* Standby duty must not be mixed with other duty between off duty periods (FLIGHT DECK)
* Minimum days between standby (FLIGHT DECK, functional)
* Minimum days between standby and long off-duty period (FLIGHT DECK, functional)
* Max standby blocks in planning period (FLIGHT DECK, functional)
* Minimum days between standby and standby line for long-haul crew (FLIGHT DECK, functional)

### Penalties

Penalties that apply to standby are (see [6.3.3], and [6.3.4]):

* Penalty for each unassigned standby block day
* Penalty for stepping over a 7x24 duty time buffer with additional duty time for standby blocks
* Penalty for having less than 3 days off before and after a long standby duty
* Penalty for assigning a standby directly before a free weekend
* Penalty for assigning a standby block too close to the previous standby line
* Penalty for assigning a standby line too close to the previous standby line

### Constraints

Constraints that apply to standby are (See [6.4.1] and [6.4.2]):

* Maximum standby FTE assignment per day, rank, base, start time
* Minimum standby FTE assignment per day, rank, base, start time
* Max number of single qualified crew per day with standby line assignment

### Standby optimization

Standby blocks are regarded as normal trips in the optimization in the sense that they have an assign value that should be fully covered. The optimizer will, guided by the cost function, try to assign as many standby blocks as possible.

Standby line assignment is also done by the optimizer but lines need not be fully assigned. Instead there are vertical constraints that tries to push the assigned number of different standby lines per day towards a target expressed in the table *standby\_line\_need* (see [8.1.5.1]). There is no cost of unassigned standby lines, but through the vertical constraints there is a cost of assigned standby lines if the assigned value differs from the target value (see [6.4.1]). Standby lines should be created with a booking value higher than the target, as to not prevent the optimizer from finding the best overall solution.

There are two constraints for each target in the table *standby\_line\_need*, one lower and one upper constraint. Typically the lower constraint is always turned on with a higher cost than the upper constraint to make it more costly to not reach the target than to overshoot. The upper constraint should be used if standby lines are optimized separately and there is no other pressure to keep the assigned amount down. If the standby lines are included with the normal production, the upper constraint could be turned off and the cost functions tuned so that a balance between assigned standby lines and unassigned production is achieved.

Apart from the constraints described above, there is also a constraint that can be used to limit the number of single qualified crew per day that are assigned standby. The max limit is set using a parameter and single qualified crew with standby will be counted, regardless of base, rank or start-time of the standby.

Scattered standby are not handled by the optimizer. If needed, they have to be manually assigned post optimization.

Standby block and standby line trips must be manually created according to the standby need of the coming planning period with respect to number of crew, aircraft qualifications, crew functions and type of standby. This can be done directly in the plan on which optimization will be used or in a separate plan that is later on fetched before starting the optimization runs.

#### User configuration

The table *standby\_line\_need* is used to configure the assignment of standby lines. Standby line need can be specified per base, rank, start time (of day) and date. For base a wildcard ‘\*’ is allowed. Constraints can be turned on or off individually, either by changing the date fields or the activation field.

Each row in the Table [3] corresponds to three vertical constraints, one that pushes the assigned value up towards the target and one that keeps the assigned value from growing too much above the target. The third constraint spreads the starts of the SB:s so to avoid multiple starts in a short interval. The three constraints can be turned on or off individually and the cost of breaking the constraint can be set separately.

**Example:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Base | Rank | Start time | Start end | Validfrom | Validto | Target | Active | Spread | Spread target | Overlap days | Double qualified |
| \* | FC | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 300 | False | 96:00 | 1 | 0 | False |
| \* | FC | 12:00 | 12:00 | 01JAN1986 | 31DEC2099 | 300 | False | 96:00 | 1 | 0 | False |
| \* | FP | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 300 | False | 96:00 | 1 | 0 | False |
| \* | FP | 12:00 | 12:00 | 01JAN1986 | 31DEC2099 | 300 | False | 96:00 | 1 | 0 | False |
| STO | AP | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 200 | False | 96:00 | 1 | 0 | False |
| STO | AS | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 400 | False | 96:00 | 1 | 0 | False |
| STO | AH | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 1200 | False | 96:00 | 1 | 0 | False |
| OSL | AP | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 500 | True | 96:00 | 1 | 0 | False |
| OSL | AS | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 800 | True | 96:00 | 1 | 0 | False |
| OSL | AH | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 1500 | True | 96:00 | 1 | 0 | False |
| CPH | AP | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 300 | False | 96:00 | 1 | 0 | False |
| CPH | AP | 12:00 | 12:00 | 01JAN1986 | 31DEC2099 | 300 | False | 96:00 | 1 | 0 | False |
| CPH | AS | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 400 | False | 96:00 | 1 | 0 | False |
| CPH | AS | 12:00 | 12:00 | 01JAN1986 | 31DEC2099 | 400 | False | 96:00 | 1 | 0 | False |
| CPH | AH | 06:00 | 06:00 | 01JAN1986 | 31DEC2099 | 1500 | False | 96:00 | 1 | 0 | False |
| CPH | AH | 12:00 | 12:00 | 01JAN1986 | 31DEC2099 | 1500 | False | 96:00 | 1 | 0 | False |

1. standby\_line\_need

Fields:

* Base: The base for which the constraint applies. Can be a wildcard, ‘\*’, which means that the constraints cover all bases. I.e. the first row in the table above says that the optimal need value of standby lines starting 06:00 for all crew with rank FC regardless of base is 3.
* Rank: The rank for which the constraint applies.
* Starttime: The start time of day of the standby line. Used to distinguish between early and late standby etc.
* Startend: If two standby lines are considered equal a interval can be used and as long as the starttime is within the interval it is considered in the constraint. If start time is 6:00 and and start end is 8:00 every SB line starting between 6 and 8 is considered.
* Validfrom: The first day this constraint is valid.
* Validto: The last day this constraint is valid.
* Target: target for standby line assignment that the constraint is aiming to achieve. The target is expressed in hundredths of a need. This is interpreted differently for FD and Cabin, for Cabin the need is interpreted as The full time equivalent (F.T.E.), For FD F.T.P full time producing as during a standby line every crew is considered the same.
* Active: If True this constraint is active and will be recognized by the optimizer
* Spread: Defines the width of the window that will be considered for the spreading of SB-line starts. The size of the window usually depends of the target, a high target usually means a small spread window, and the other way around. A spread of 96 = 4\*24:00 means that it is ok to have a new sb line starting after 4 days without having a penalty.
* Spread target: Defines how many starts is ok to have with in the spread time window. Normally this should always be one. But if the target is large, a higher number might be necessary. Say we have a target of 15 and the lines are 12 days long. The spread is 24:00 it is not possible to fulfil with have a sb line starting once every day, we sometimes need two. In this case a higher spread target is recommended, in this case 2 would be good.
* Overlap: Means that the SB-lines should overlap, this might be necessary for very thin problems. Say that the target is 1, this means that there will only be 1 SB line for the entire period. This means that the standby will not be able to cover trips over the joint from one sb line to the other, but if those two sb-lines would overlap even such trips will be covered.
* Double qualified: Only used for flight deck. True means that only double qualified crew will be considered for this SB-line. This is again for standby line with a low target, as a line now will cover production for two types(A340/A330). The alternative would be that we could only recover trips from one of those types which would not be ideal, if there is a disturbance for the other one.

As few constraints as possible should be active; generally only the constraints relevant to the area with which the planner is working should be active. An active constraint for rank AP will not influence the cost of the solution when planning a flight crew area, but the optimizer will always evaluate it and hence waste time. Best practice is to keep a local copy of the table in the sub-plan and rows not matching anything in planning problem at hand.

### Planning tools

There are various tools that support the standby assignment such as reports, filters and sorting.

#### Reports

The following Standby reports are available and documented in the ‘Functional Reference – Reports’.

* **Standby Distribution** This report displays distribution of assigned short haul standby over the planning period
* **Standby Line Distribution** This report displays information on the active standby line constraints and the outcome of the current assignment compared to the targets

#### Sort

Crew can be sorted by the number of standby days in the period, previous period and year, the first in the period or the last standby crew have had. The sorts are accessible from the “Assignment General” menu.

## Blank days

The function for assigning blank days to crew is available from “Create > Blank days”. It will try and assign blank days on the roster with the following priorities:

* + - 1. Fill holes inside wops.
      2. Add blank days after wops.
      3. Add blank days before wops.
      4. Add blank days on all open days on the roster (starting from a random offset to avoid having a lot of blank days assigned in the beginning of the planning period.

All of the assignments are done legally, i.e. no assigned blank day will make the roster illegal. If the crew is illegal initially, no blank days will be assigned. This introduces a difficulty; since there are separate rules that control the amount of blank days on the roster, it is entirely possible that the roster will have a lot of open time where blank days cannot be assigned, but where production is possible. To circumvent this one can turn of blank day-related rules before using the assignment function. This will enable more realistic values in the Balance Distribution report.

### Rules

The rules handling blank days are (see the Rule Overview document for details):

* Max BL-days in calendar year (CABIN CREW SKD)
* Max BL-days in calendar month (CABIN CREW SKS)
* No BL-day for very inexperienced crew (CABIN CREW, functional)
* No BL-day day before BBK trip (CABIN CREW, functional)
* No single empty day after BL-day (FLIGHT DECK, functional)
* Max BL-days in planning period (FLIGHT DECK, functional)

### Penalties

Penalties that apply to blank days are (see [6.3.3]):

* Penalty for each unassigned blank day
* Penalty for stepping over a 7x24 duty time buffer with additional duty time for blank days
* Penalty for assigning a blank day first in a working period
* Penalty for assigning a blank day after a late checkout
* Penalty for assigning a blank day before an early checkin
* Penalty for assigning a blank day before a free weekend
* Award for assigning a blank day after a standby
* Award for assigning blank days adjacent to each other
* Penalty for “unfair” distribution of blank days

### Blank day optimization

Just as with standbys, a blank day trips of varying length and crew need must be manually created before the planning period. This can be done directly in the plan on which optimization will be used or in a separate plan that is later on fetched before starting the optimization runs.

Blank days are generally cheap to leave unassigned, and there is no need to assign everything. Thus the blank day input can, and probably should, contain more blank days than needed.

### Planning tools

There are various tools that support the blank day assignment such as reports, filters and sorting.

#### Reports

The following blank day reports are available and documented in the ‘Functional Reference – Reports’.

* **Blankday Distribution** This report displays distribution of assigned blank days over the planning period (See Functional Reference – Reports for details)

#### Sort

Crew can be sorted by the number of blank days in the period and the first blank day in the period. The sorts are accessible from the “Assignment General” menu

## Compensation days (bid)

The function will try to assign crew’s compensation day bids in the planning period. Assignment uses the defined bid-table in ruleset. The assignment of compday uses, like blankdays, legality, thus no illegal bids will be fulfilled. A result of this is that roster needs to be legal before assignment starts; otherwise script cannot assign any bid for this crew.

The compdays will check balance against crew account.

## Freedays

The functions for assigning freedays are available from **Create > Freedays** on the Assignment general menu

### Pre-optimization freedays (fixed freedays)

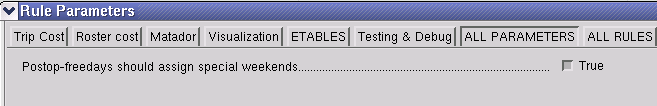
The function will try to assign freedays to all crew in window that have a fixed-group contract for at least part of the planning period. The function will assign all daytypes in the pattern except P and FP, for all days specified. The daytype for a specific date is retrieved after checking the contract at that date, so contract changes are handled automatically. A special case is when a crew changes from variable group to fixed group. In that case two freedays are assigned the two days before the fixed pattern starts.

An error message (listing crew by employee number) will be shown if a freeday assignment fails. This will typically happen because of overlapping trips (only overlapping on duty activities will result in error messages), in which case the trip code and date will be shown. Broken or otherwise faulty data will typically result in an “Unknown reason” message.

### Post-optimization freedays

The function assigning freedays post-optimization (after an optimizer run) will assign:

* F4-days on all valid free weekends.
* Special weekends. The assignment is contolled by parameter and can be turned of for cabin.



1. Assign special weekends control.

* Carry-out F-days for long trips with required F days into the next planning period (two freedays are assigned on trips with required rest where the rest period extends into the next planning period).
* Flex group F-days (days with code “FP” in the pattern, for 5/4 Flex crew).
* Part time freedays and special schedule part time ill freedays according to requirement.
* Required freedays before vacation.
* Last, it will assign required freedays after wop and then add remaining needed freedays (for monthly requirements).

### Fill roster with freedays

The last step before Publish is to make sure there are no holes on the roster. This function will assign freedays wherever possible, while still not creating any single days off.

## Special Schedules

In the Crew Info interface a tab for Special Schedules is available. It provides a simplified interface to the etable for special schedules. A new item is created by setting the required keys (Valid from, Type and Note) to unique values. After an item is created the other fields can be changed. The key fields, however, can not be changed (the work-around is to delete the item and create it as new).

The fields have different functions for different item types. Below is a table describing which fields are used for what type.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Type | Valid from | Valid to | Note | From | To | Time |
| ForbiddenDest | x | x | x | - | - | - |
| ForbiddenAct | x | x | x | - | - | - |
| TripLength | x | x | - | x | x | - |
| TimeOff | x | x | - | x | x | - |
| CheckIn | x | x | - | x | x | x |
| CheckOut | x | x | - | x | x | x |
| MaxDuty | x | x | - | - | - | x |
| MaxLegs | x | x | - | - | x | - |
| MaxBlh | x | x | - | - | - | x |
| PartTime | x | x | - | - | x | - |

The unused types can be left empty.

The fields "From” and "To" have different meaning depending on type:

* For types "TimeOff", "CheckIn" and "CheckOut", the valid weekdays interval should be entered (e.g. "TimeOff every Friday" will be entered as 5, 5 and "CheckIn later than 07:00 every weekend" will be entered as 6, 7, 07:00).
* For type "TripLength", the valid trip length interval should be entered (e.g. "Only 2 and 3 day trips allowed" will be entered as 2, 3).

The field “Time” also have different meaning depending on type:

* For types “CheckIn” and “CheckOut” the time is the time-of-day of the limit (e.g. check-in before a certain time of day and check-out after a certain time-of-day could be forbidden).
* For types “MaxDuty” and “MaxBlh” the time is the maximum number of hours for duty and block hours, respectively.

The field “Note” can be used to specify:

* Forbidden destination (“ForbiddenDest”)
* Forbidden activity (“ForbiddenAct”).   
  “ForbiddenAct” can take both a group code and a task code, depending on how broad or detailed the prohibition is to be. The table *activity\_set* is the base for group code.
* Forbidden AC Family. The families are:
  + - A320
    - A330
    - A340
    - B737
    - CRJ
    - F50
    - MD80
* Forbidden AC Family can also be used to forbid certain aircraft qual, e.g.   
  Note=’38’ will forbid the acqual 38 but not acqual 37. Note=’B737’ will forbid them both.

## Crew Warnings

The crew warnings package is used to assist the planners by visualizing important aspects in a schedule that must be taken care of but is not illegal. A crew member might for example not be allowed to fly next month if he or she is not assigned the proper training the current month.

Prioritized warnings are visualized in the crew margin. There is also a report that can be used to display all warnings for a crew of group of crew and the possibility to filter crew with certain warnings. All crew except SKN cabin crew with only short haul qualifications can have warnings.

Warnings can have a priority, where upper case takes precedence over lower case. ‘S’ has higher priority than ‘s’ but not ‘y’, if both ‘S’ and ‘s’ applies then only ‘S’ will be displayed.

The absence warnings (4/A for flight crew and a/A/f/F for cabin crew) use the same mechanisms as the corresponding legality rules, in order to get the full details the crew warning report must be used. The report displays the same information as the corresponding legality rules when it comes to training need.

### Flight Deck Warnings

These are the warnings for flight crew:

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Prio | Displayed when | Description |
| S | 1 | OPC date <= end of current month and no OPC planned during the current month | OPC needed this month and the OPC expiry date. Ex: *OPC needed current month, expiry date is 01Jul2007* |
| s | 2 | End of this month < OPC date <= end of next month and no OPC planned during the next month | OPC needed next month and the OPC expiry date. Ex: *OPC needed next month, expiry date is 01Jul2007* |
| Y | 1 | PC date <= end of current month and no PC planned during the current month | PC needed this month and the PC expiry date. Ex: *PC needed current month, expiry date is 01Jul2007* |
| y | 2 | End of this month < PC date <= end of next month and no PC planned during the next month | PC needed next month and the PC expiry date. Ex: *PC needed next month, expiry date is 01Jul2007* |
| E | 1 | PGT date <= end of current month and no PGT planned during the current month | PGT needed this month and the PGT expiry date. Ex: PGT *needed current month, expiry date is 01Jul2007* |
| e | 2 | End of this month < PGT date <= end of next month and no PGT planned during the next month | PGT needed next month and the PGT expiry date. Ex: *PGT needed next month, expiry date is 01Jul2007* |
| C | 1 | LC date <= end of current month and no LC planned during the current month | LC needed this month and the LC expiry date. Ex: Line check *needed current month, expiry date is 01Jul2007* |
| c | 2 | End of this month < LC date <= end of next month and no LC planned during the next month | LC needed next month and the LC expiry date. Ex: *Line check needed next month, expiry date is 01Jul2007* |
| 4 | 1 | Crew is absent from flight duty 46-90 days at the start of the planning period or will become absent from flight duty 46-90 days during the planning period and no flight production is assigned after the absent date | Crew is absent from all flight duty 46-90 days, what training is needed and what date crew entered the particular absence interval. Ex: *Absence M8 46-90 days, needs SIM and 2 legs with non-restr or SIM (09Apr2007)* |
| A | 1 | Crew is absent from flight duty > 90 days at the start of the planning period or will become absent from flight duty > 90 days during the planning period and no flight production is assigned after the absent date | Crew is absent from all flight duty >90 days, what training is needed and what date crew entered the particular absence interval. Ex: *Absence M8 91-180 days, needs SIM and 2 T- or LC-flts (24May2007)* |
| T | 1 | Last recent date <= end of the planning period no flight production is assigned after the last recent date | Crew lost recency, what training is needed and last recent date. Ex: *Last recent date is 23May2007, will need T-FLT or SIM* |
| D |  | SKI Flight Deck airport/area expiry date <= end of this month and no production for that airport is assigned after the expiry date | Which airports are due and at which date crew last flew there. If the airport corresponds to a region with requirements, that is also displayed. Ex: *Airport qualification(s) expire current month: EWR (US) (01Jan1987), IAD (US) (01Jan1987), ORD (US) (01Jan1987), SEA (US) (01Jan1987)* |
| d |  | SKI Flight Deck airport/area expiry date <= end of next month and no production for that airport is assigned after the expiry date | Which airports are due and at which date crew last flew there. If the airport corresponds to a region with requirements, that is also displayed. Ex: *Airport qualification(s) expire next month: EWR (US) (01Jan1987), IAD (US) (01Jan1987), ORD (US) (01Jan1987), SEA (US) (01Jan1987)* |
| L | 1 | SKI crew with LCP qualification has not flown, and is not planned to, a SKI destination during 1 year before the end of the current month | Which airport(s) are due current month and which date they were last flown. Ex: *Airport due current month: PEK (01Jan1987)* |
| l | 2 | SKI crew with LCP qualification has not flown, and is not planned to, each SKI destination during 1 year before the end of next month | Which airport(s) are due next month and which date they were last flown. Ex: *Airport due next month: PEK (01Jan1987)* |
| R | 1 | Crew is missing registration of a recurrent date at the start of the planning period | Which recurrent dates are missing. Ex: *Recurrent OPCA4, PCA4, PGT dates must be registered* |
| v | 2 | A simulator is assigned too close after vacation (parameter setting) | Ex: *OPC/PC planned too close after vacation* |
| U | 1 | Crew is on training at start of the planning period (crew\_training\_need) and has not yet performed all training | Crew has not performed all required training, what type of training is needed next according to the training program and what qualification (if any) is relevant. |

### Cabin Crew Warnings

These are the warnings for cabin crew (NB: Not applicable to SKN cabin crew with only short haul qualifications):

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Prio | Displayed when | Information given in report CrewWarningInfo: |
| H | 1 | The registered recurrent date is less than start of planning period or no recurrent date is found at all | Previous years recurrent training has not been performed and the date it should have been performed or if it is missing entirely. Ex: *Missing previous year's recurrent training, expiry date was not found* |
| E | 1 | Recurrent date <= end of current month and no recurrent training planned during the current month | Recurrent training needed this month and the recurrent expiry date. Ex: *Recurrent training needed current month, expiry date is 01Jul2007* |
| e | 2 | End of this month < Recurrent date <= end of next month and no recurrent training planned during the next month | Recurrent training needed next month and the recurrent expiry date. Ex: *Recurrent training needed next month, expiry date is 01Jul2007* |
| U | 1 | Crew is on training at start of the planning period (crew\_training\_need) and has not yet performed all training | Crew has not performed all required training, what type of training is needed next according to the training program and what qualification (if any) is relevant. Ex: *Crew on training REF 6-12-A2AL, missing 4 SUPERNUMERY (A2)* |
| A | 1 | Crew is absent from ac type > 90 days at the start of the planning period or will become absent from ac type > 90 days during the planning period and no flight duty has been assigned after the absent date. | Crew is absent from a particular ac type, what training is needed and what date crew entered the particular absence interval. Ex: *Absence 90 91-365 days, needs REF and 2 RELEASE (90) in 14 days (27May2007)* |
| F | 1 | Crew is absent from flight duty > 90 days at the start of the planning period or will become absent from flight duty > 90 days during the planning period and no flight duty has been assigned after the absent date. | Crew is absent from all flight duty, what training is needed and what date crew entered the particular absence interval. Ex: *Absence all 91-180 days, needs REF and 2 RELEASE () in 14 days (23May2007)* |

### Reports

The following crew warning report is available and documented in the ‘Functional Reference – Reports’.

* **Crew Warning Info** This report displays detailed warnings for a single crew or for crew in window (See Functional Reference – Reports for details)

### Filters

There are filters for finding crew with any warnings or crew with a specific warning; these are found on the Assignment general menu.

### Visual indicators

A concatenated string of all warnings are displayed in the left margin in the crew window, lower right area. This can be turned off with a parameter in the Visualization tab in the parameter form.

**Example:** Cabin Crew

CC_Warning_margin2

**Example:** Flight Deck

FD_Warning_margin2

## Rule Exceptions

Rule exceptions enable you to make the legality rules less restrictive, or completely turned off, for individual crewmembers at given occasions. A rule exception works in two ways, depending on how the rule is written.

### Extension

Most rules have the following structure:

value <= limit (or value >= limit)

An example is: Duty time in 7x24 hours <= 42 hours.

An exception to a rule of this type extends the limit during the rule’s scope of validity, making the rule legal. In the example the scope of validity is the 7x24 hours where the rule is violated, not any 7x24 hours within the roster.

### Turn Off

If the construction of the rule does not allow a limit extension, the rule exception completely turns off the rule during the rule’s scope of validity.

### Usage

The instructions here mostly apply when working with file plans. When working with database plans there are other means of creating and removing rule exceptions.

#### Investigate rule violations

It is important to make sure that all crewmembers have legal rosters. Sometimes there are rule violations that are acceptable / inevitable and should be overridden, but they should be investigated and confirmed before the rule violation can be waived.

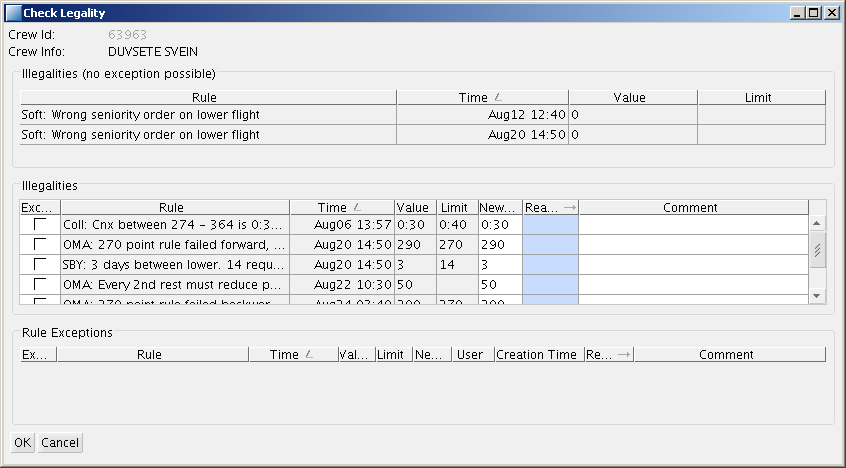
To filter and investigate the rule violations do the following:

1. Show rosters in window 1.
2. Use the command **Window menu >Show Illegal/Legal >Show Illegal Rosters**

This displays all illegal rosters and as mentioned above they should be investigated since they indicate that something could be wrong in the feeds or in the rules. There could also be deliberately made rule violations, and then they should explicitly be waived.

#### Make rule exception in database plan

When working in a database plan rule-exceptions are set on crew on failure by failure basis. It is also possible set the new limit manually. By using this, an illegality can be relaxed even more than the current failure. In a database plan all rule exception are handled on individual crew basis. Use the command **Crew Object > Rule Exceptions > Edit** to add or remove rule exceptions using the dialog shown in Figure [16].





#### Make rule exception in file plan

To waive the rule violations on all crew displayed in the window, use the command **Crew General > Rule Exceptions > Create** to ignore/override the rule violations and being able to assign additional tasks to these rosters, or to start optimization runs.

If the rule exception should only be made on one individual crew member use the command **Crew Object > Rule Exceptions > Create** instead.

If you only want to make Rule Exception’s on some rules, the others need to be turned off while the exceptions are made. When the rules are turned back on the roster will appear illegal.

#### Remove rule exception in file plan

It is also possible to remove a rule exception. Use the command **Crew General > Rule Exceptions > Clear** to remove the rule exception and the crewmember becomes illegal again.

**Note:** If the rule exception should only be removed from one individual crew member use the command **Crew Object > Rule Exceptions > Clear** instead.

### Reports

The following rule exception report is available and documented in the ‘Functional Reference – Reports’.

* **Rule Exception Info** This report displays detailed rule exceptions for a single crew or for crew in window (See Functional Reference – Reports for details)

Apart from the normal report menu entries it is also directly available using the **Crew Object > Rule Exceptions > View** and **Assignment General > Rule Exceptions > View** commands.

### Visual indicators

A crewmember having a rule exception is visualised by a brown marker in the crew header. A trip that is causing a rule exception is visualised by a brown line underneath the trip.

RuleException2

# GUI

## Filters

All filters are defined at the start of the planning period, unless otherwise noted.

Menu items ending with “…” (three dots) brings up a form where more options are available.

Menu items ending with “>” (an arrow-like shape) have a sub-menu.

All standard filters use the planning area as an added criterion.

The filters available on the Subfilter menu work in the same way as the corresponding standard filter, except that they don’t use the planning area.

Crew window:

* **by Filtering Mask…**Opens the main filtering form with a large number of options
* **by Surname/Empno/ID…**Opens a smaller filtering form where crew can be filtered by surname, employee number or crew id. Comma-separated values are allowed to enable filtering of multiple crew.
* **Mini Filter…**Opens a smaller filter form with various more advanced options
* **Rank >**
  + **Flight Deck/Cabin crew**Filters all flight deck crew/all cabin crew
  + **FC, FP, FR/AP, AS, AH**Filters all crew with specified rank
* **Region >**
  + **SKD/SKN/SKS/SKL/SKI**Filters all crew with the specified region
* **Homebase**
  + **CPH/OSL/STO/…**Filters all crew with the specified home base
* **Contract >**
  + **Full time/Part time**Filters all full time/part time working crew
  + **Variable group/Fixed group**Filters all crew in variable group/fixed group
  + **Fixed group (excl. specials)**Filters fixed group crew excluding 5/4 flex and F/V group
  + **5/4 Flex**Filters all crew with a 5/4 flex contract (SKN)
  + **F/V Group**Filters all crew with a fixed-variable group contract (SKN)
  + **Resource Pool**Filters all resource pool crew (SKN)
  + **Passive +**

Filters all crew in group Passive +

* **Qualification >**
  + **Aircraft >**
    - **Longhaul**Filters all crew with long haul qualification
    - **Shorthaul only**Filters all crew without long haul qualification
    - Common
      * **36, 37, 38, ...**  
        Filters all crew with specified aircraft qualification
    - Flight Deck specific
      * **A3, A4, M8**Filters all crew with specified flight crew related aircraft qualification
    - Cabin Crew specific
      * **AL, 90**  
        Filters all crew with specified cabin crew related aircraft qualification
  + **Airport >**
    - **US (EWR, IAD, ORD, SEA)**Filters all crew qualified for US airports
    - **FNC, HMV, …**Filters all crew with specified airport qualification
  + **Airport, expired**  
    Filters crew who have had the specified qualification, but that is expired
* **Restrictions >**
  + **Any**Filters crew with any restriction
  + **Any new**Filters crew with any restriction of type “NEW”
  + **New on rank**
  + Filters crew that are new on rank
  + **New employee**Filters crew that are new in company (NEW+NEW)
  + **New on A/C type**Filters crew that are new on AC type
  + **New 6 months**Filters crew that are NEW+6M
  + **Any medical**Filters crew with any medical restriction
  + **Medical (medic)**Filters crew with any medical/medic restriction
  + **Medical (60+)**Filters crew that have passed 60 years of age in the planning period
  + **Training Capt**Filters crew that are training to become captain
  + **Training DCT**Filters crew during conversion training
* **Crew warnings**

Please see 8.6 for a description of the warnings

* **Misc >**
  + **Annotations…**Opens a form to filter crew with annotations
  + **Special schedules**Filters crew with special schedule
  + **Standbys**
    - **Standby blocks**Filters crew with standby blocks
    - **Standby lines**Filters crew with standby lines
    - **Scattered standby**Filters crew with scattered standby
  + **Crew with off duties whole planning period**Filters crew with no production and no available days
  + **Crew with holes in publ. period.**   
    Filters crew with holes in roster. Used before publish to manually handle empty periods before publish fills them with freedays. The list will be sorted with crewmember with most empty gaps on top.
  + **Crew with overlaps in publ. period.**   
    Will display crew with overlaps. Used before publish.
* **Lifestyles >**
  + **Have Lifestyle Bid**Filters crew with lifestyle bids
  + **Lifestyles..**
    - **Nights Away from Home Base**
    - **Nights at home**
    - **Morning Person**
    - **Evening Person**
    - **West Destinations**
    - **East Destinations**
* **Bids >**
  + **Bids in pp**Filters crew with bids in period
  + **Bid ratio**
    - **Above target**Filters crew with a bid ratio at or above target
    - **Below target**Filters crew with a bid ratio below target
  + **Compensation days**
    - **Compday bids in pp**Filters crew with compensation day bids
    - **Unfulfilled compday bids in pp**As previous filter, but only for bids that haven’t been fulfilled
    - **High balance**Filters crew that can have compensation days forced
* **Instructors >**
  + **Flight Deck**
    - **All (LIFUS, TRI/TRE, SFI/SFE)**Filters all crew with an instructor qualification, except OPT and CRM
    - **LIFUS**Filters all crew with a LIFUS instructor qualification
    - **FC LIFUS**All FC LIFUS instructors
    - **FP/FR Instructors**All FP/FR crew with instructor qualification
    - **LCP**Filters all crew with an LCP instructor qualification
    - **TRI/TRE**Filters all crew with a TRI or TRE instructor qualification
    - **SFI/SFE**Filters all crew with an SFI or SFE instructor qualification
    - **OPT/CRM**Filters all crew with an OPT or CRM instructor qualification
    - **SUP**Filters crew with a SUP (Supervision) qualification
  + **Cabin Crew**Filters all crew with the specified instructor qualification
* **Recurrent > These are documented in Functional Reference Training**
* **Training > These are documented in Functional Reference Training**
* **Recency >**
  + **Unrecent pp-start**Filters crew that are unrecent at the start of the planning period
  + **Unrecent pp-end**Filters crew that are unrecent at the end of the planning period
* **Outside Planning Area**Filters crew not in the currently defined planning area
* **Optimizer input**Filters crew that were input to an optimization

Trip window (all filters available in Pairing is documented in Functional Reference Pairing)

* **Manko in planning period** – Filters uncovered production in the planning period
* **Training**
  + **Assist OPC/PC** – Simulators in need of assist (i.e. missing one pilot)
* **Extra seats overbooked** – Trips with extra seat legs that are booked above available cabin seats

## Filter variables

Variables that can be used in the filter form where they should be input in the rule values section and matched against the desired value.

*Filter flight crew with a given PC or OPC month*

The variable should be set equal to the month number, e.g. 1 for January and 12 for December. To find crew that lacks a recurrent month match the variable against 0. The variables that end with *\_dq* are for filtering the months for double qualified crew.

* **studio\_select.pc\_month**
* **studio\_select.opc\_month**
* **studio\_select.pc\_month\_dq**
* **studio\_select.opc\_month\_dq**

*Filter flight crew with a given PC or OPC year*

The variable should be set equal to the year, e.g. 2007 or 2008. To find crew that lacks a recurrent year match the variable against 0. The variables that end with *\_dq* are for filtering the year for double qualified crew.

* **studio\_select.pc\_year**
* **studio\_select.opc\_year**
* **studio\_select.pc\_year\_dq**
* **studio\_select.opc\_year\_dq**

## Sorts

Trip window:

* **Departure time**
* **Arrival time**
* **Departure airport**
* **Arrival airport**
* **Crew complement**Sorts trips according to number of available positions
* **Rule values**Displays a form where a rave variable can be entered, to sort trips by
* **Cost**Sorts trips according to current trip costs
* **Duty time**

Crew window

* **Crew Alias**
* **Rule Values...**Displays a form where a rave variable can be entered, to sort crew by
* **Employee number**
* **Home base**
* **Group**Sorts crew by variable or fixed group, incl. pattern and cycle start
* **Parttime factor**
* **Freeday balance**Sorts crew according to extra number of freedays assigned
* **Seniority**
* **Cost**Sorts crew according to current roster costs
* **Lifestyle Fulfillment**
* **Bid Points**
* **Last flown**
  + **ALL**Last flown on any ac type
  + **Any 737**Last flown on 36, 37, or 38
  + **A2, CJ, F5...**Last flown on specified type
* **Recurrent**
  + **PC/OPC month**First of PC or OPC month
  + **PC, OPC, PGT, LC, REC month**Sorts crew according to the month number of the specified recurrent type
  + **PGT occurrences**Sorts crew according to first PGT occurrence after start of planning period
  + **Sim occurrences**Sorts crew according to first simulator occurrence after start of planning period
* **Blank day**
  + **Number in period**Sorts crew according to number of blank days, with crew with no blank days on top
  + **First in period**Sorts crew according to the first blank day occurrence in the planning period
* **Standby**
  + **Number in period**Sorts crew according to number of standby days and lines in the planning period
  + **Number in previous period**Sorts crew according to number of standby days and lines in the month before the planning period
  + **Number in year**Sorts crew according to number of standby days and lines in the current calendar year
  + **First in period**Sorts crew according to the first standby occurrence in the planning period
  + **Last (incl. period)**Sorts crew according to the last standby occurrence on the roster

## Find

* Find comrades on leg (for unassigned trip object and assignment object (i.e. assigned trip).

## Temporary filtering

It is possible to temporarily store the filtering of crew, currently in the window, for later use. This is convenient of for example a set of instructors have been filtered in the window, and another filtering has to be made for else, and then the same instructors should be filtered again. In this case a temporary filtering could be created from the instructors in the window, the new filtering could be made, and then the instructors could easily be retrieved by using the previously stored temporary filtering.

The temporary filtering is unique for a user, it is not possible to use a temporary filter created by someone else. The existing filters can be inspected using **Planning Tools -> Table Editor -> File… -> Temporary Tables**.

A temporary filter is created using the command **Assignment General -> Filter -> Save to temporary filtering -> [1-5].** At most 5 filters can be stored. There is no information added to the menus of what filters consists of what, that has to be noted by the user.

A temporary filter is used with the command **Assignment General -> Filter -> Use temporary filtering -> [1-5]**,or as a sub-filter using **Assignment General -> Subfilter -> Use temporary filtering -> [1-5]**.

## Predefined filters

A result of a temporary filtering, or any list of crew id-s, can be used to create a predefined filter.

The existing predefined filters are managed using **Planning Tools -> Table Editor -> File… -> Predefined filtering**. A new predefined filter can for example be created by copying, and renaming, a temporary filter. Predefined filtering should be used sparingly since they are global and available to all users.

The filters are available using **Assignment General -> Filter -> Use predefined filtering -> <name of filter table>** and **Assignment General -> Subfilter -> Use predefined filtering -> <name of filter table>**

# Overlapping Activities

When working with live data updated both through external interfaces and manual transactions there is always an inherent risk of data conflicts. One such conflict is when legs overlap each other in time on the roster, or in open trips.

In a rostering system where there is an online environment updated both through external interfaces and manual transactions there is an inherent risk of overlapping activities. There might be overlaps due to delays over freedays or other flights etc. Things move and what looks ok one minute might have changed the next.

At any time a rostering planner might export a file based rostering scenario for running Matador. However, Matador cannot handle overlapping activities and will not start if presented with such input.

## Resolving overlaps

How the overlaps are to be handled is up to the experience of the planner, they know best what the likely outcome of an overlap would be. Depending on which legs are kept on/removed from the rosters the optimization results might be affected in different ways, if legs important for recency or absence legality are removed crew might not get any assignments. Note that this might be the actual, and desired, outcome, as stated earlier the experience of the planner is important in this step.

The system can help the planner by guessing which legs to remove to clear each overlap. The guess is based on the following algorithm:

1. Non-flight legs are chosen before flight legs.
2. If legs of same kind (flight/flight or non-flight/non-flight) chose the shortest.
3. If legs in (2) are of the same length, chose the latest.

### Filtering and viewing

The following commands are available to filter and view overlaps:

#### Command: Assignment General -> Overlaps -> Filter

Filters crew with overlapping activities and marks all legs involved in an overlap, if two legs overlap each other both are marked.

#### Command: Assignment General -> Overlaps -> Report

This command will display a text report listing all the overlapping activities. The activities that the best guess algorithm will keep are marked with an asterisk (\*).

### Removing

After using any of the filtering commands listed in [10.1.1] these commands it is then possible to browse through the filtered rosters, inspect the overlaps and adjust the guesses made. It is then possible to remove the marked legs using the standard functionality.

It is also possible to let the system automatically remove overlaps, these are the commands:

#### Command: Assignment General -> Overlaps -> Remove

For each overlap it will remove the legs most likely for removal according to the algorithm above. Only available in file plan to enable optimization. In database overlaps must be manually inspected and removed.

# Publish

Each month the 15th the next month’s rosters are released to crew, and the roster maintenance and crew tracking department assumes the responsibility of the rosters. The roster publication functionality from a planning perspective is described here. The publication and re-scheduling in the maintenance phase is described elsewhere.

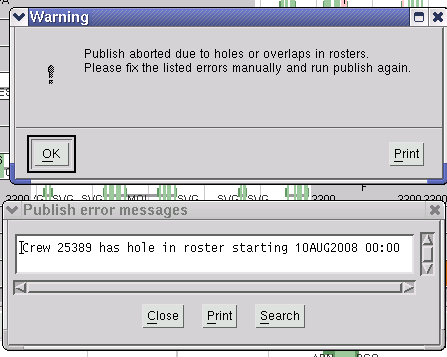
During the roster publish a few different steps are performed:

* Any rostered compensation day in the publication month is marked as published.
* All data needed from the published rosters in order for legality to work in next period is calculated and stored in the so-called accumulators.
* Data needed for re-scheduling rules during “disponering” and tracking is stored.
* All assigned non-personal standby and blank day duties in the publication period of the roster are replaced by one-day personal activities on the roster. This is because tracking can not work with these objects in a correct way if they are not personal.
* The actual rosters are marked published for the period, i.e. a month by default, but for crew changing area in the month it will be the period they are in the published planning area. This means that the next time crew accesses the crew portal they will see this updated version of their roster in the publication period.

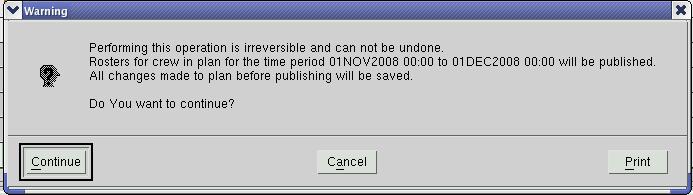
## Basic publishing step

The basic publishing step is performed as follows:

1. Open the production database for the planning area that is to be published
2. Set the correct planning period. The publishing period will be the month that includes the start of the planning period.
3. Make sure the rosters are ready for publication and make last minute adjustments
4. Select the menu option **Planning Tools -> Publish plan**
5. It is not possible to publish plan if there exists empty days or overlapping activities in roster! In this case the operation is aborted and an error list and a message shown in Figure [17] is displayed.



2. The user must manually handle these cases and then publish again.  
   The filters ‘Assignment General 🡪Filter 🡪 Misc 🡪 Crew with holes in publ. period’ and ‘Assignment General 🡪Filter 🡪 Misc 🡪 Crew with overlap in publ. period’ will sort out these cases.
3. A message is displayed informing the user of that he/she is about to publish the rosters for all crew in the planning area during the selected publishing period, and that it is not reversible (see Figure [18]).



2. Confirm that it is OK to publish
   1. The user press “Continue” and the publishing step is performed
   2. The user press “Cancel” and the publishing step is aborted

The publish operation will perform a Save, which means that all other changes in other time periods of the roster, and changes to tables will be saved along with the publication status of the roster.

## Crew changing planning area in the publication period

The standard publication period is one calendar month, but for crew changing area in the publication month the period that will be published is the period where they belong to the current area (the area loaded and about to be published).

This means that the rosters for these crew will be “half published” and then “fully published” only when the other area publishes.

**Example:**

Crew ANDERSEN belongs to area SKS M8 in the publication month, in this example October, but changes to SKS B737 at the 21st of the publication month. This means that the publication period in SKS M8 will be 1 Oct – 20 Oct for ANDERSEN, and in SKS B737 it will be 21 Oct – 31 Oct.

The roster will be marked as published only for the publication period and the only the ground duties inside the period will be converted to personal activities, but the accounts, accumulators and tagging will be done for the complete month. The roster will be partially published and available to crew in the crew portal as depicted in the figure below.

The roster will be published in fragments, or steps, which means that the accumulators, accounts and tags for the whole publication month will be updated a couple of times before the roster is completely published.



1. Roster in CMS database and Crew portal before publishing the current planning month.



1. The roster as it is available to crew in the crew portal at different stages of the publishing procedure when multiple areas publish different periods of the roster in the current month. It is completely published when the last area is published.

## Conflict during Publish

In some cases the save to database that is performed within a publication might generate a conflict (see 12.3.1), which will generate a report. In most cases this happens because Tracking works on the same crew, and thereby changes some of the internal publish-related data. The publication will in that case retry up to 5 times. If it still fails to save without a conflict a message about the failure will be displayed. It is typically enough to wait a few seconds and retry the publication. If the publication finished ok the conflict reports can be safely ignored.

## Republish

If for some reason corrections have to be made to the published roster, it is possible to re-perform the publication, the previously created publication information will be updated. No re-scheduling rules are applicable in rostering so any changes between publications are allowed.

# Working with Database

## Planning period and area

The period and area chosen when opening the database will limit the amount of data loaded and available in Studio. This period and area will also be the basis for the period and area in the parameter form, once Studio is running. The settings in the parameter form can be used to further limit the planning area and period, but this will only limit what is shown, not what is available (whereas the area and period chosen when opening the database can never be expanded in the same Studio session).

Data

**visible**

in Studio

Data

in DB

Data

**available**

in Studio

**Opening**

Planning

Area/

Period

**Studio**

Planning

Area

### Exporting a file scenario

* When exporting, it’s the data available in Studio that is exported. It is for example possible to load all flight crew, then set the Studio planning area to only 737 qualified crew in STO, but the exported scenario will still have all flight crew. It is also important to note that the data loaded is dependent on period.
* An exported scenario will have its own set of all tables, frozen in time to the time of the export. This means that if an export is performed one day, and a crew gets a new qualification reported in CMS the next, the exported scenario will not know about this.
* When exporting to an already existing local plan, the etables residing in “Local plan external tables” will be overwritten. This means it is a good idea to name the local plan with the planning area and period (e.g. FC\_737\_STO\_APRIL08). This also means that if updates to for example crew data are wanted, it is possible to export a new scenario to the same local plan as a previous export, and all subplans will have the updated crew data.

## Database, Studio, Application synch

Data is stored on many levels in the system.

* The most basic is the database, where the stored data is available to all users.
* Data is loaded to Studio. Studio can get new data from the database when performing a Reload, but the database will only get data from Studio when a Save is performed.
* The applications (e.g. CrewInfo, CrewTraining, Table Editor etc.) when started from Studio will get their data from Studio. Changes in the application will not be available for Studio until Apply/Submit is performed in the application.
* When any application is started stand-alone (from the launcher) they will communicate directly with the database.

## Saving and refreshing

Since CMS is a live system, where multiple users and interfaces might modify the same data at the same time, it is important to keep the local data (loaded in Studio) up-to-date. This is done by refreshing (accessed by a double-arrow button, next to the save button). When saving, Studio will perform an internal refresh, to make sure no conflicts are created in the database.

### Conflicts and reporting

When a refresh or save generates a conflict, i.e. changes have been made in the database that conflicts with what has been done in Studio (e.g. two different planners have assigned different activities that overlap in time to the same crew), a conflict report will be shown. It will list new illegalities, what tables have been changed, and display a clickable link to show the affected rosters in a reference view.

# Fly Below Rank

The system supports crew working below their current rank in certain situations, to enable a more optimized solution. When flying below rank a duty code will be generated and displayed in the info window, and in related reports.

## Flight Deck

|  |  |  |
| --- | --- | --- |
| **Current rank** | **Assigned position** | **Duty code** |
| FC | FP | L |
| FC | FR | LL |
| FP | FR | L |

### Rules

* *(CCR) Soft: Lower rank must be scheduled in correct order*For all areas except SKS and SKD\_CJ the FC with the highest seniority (lowest value) should be assigned as FC. For SKS and SKD\_CJ the assignment should be based on the last lower date.
* *(CCR) FBR: Qualification forbidden to fly as lower rank*Certain qualifications are disallowed for lower rank assignment. These are specified in the parameter *Qualifications not allowed for below rank*. Possible values are:
  + *LCP*
  + *INSTR*: Any instructor qualification (LIFUS, TRI/TRE, SFI/SFE etc)

## Cabin Crew

|  |  |  |
| --- | --- | --- |
| **Current rank** | **Assigned position** | **Duty code** |
| AP | AH | L |
| AS | AH | L, on long haul, not considered lower on short haul |

## Reassignment script

To correct faulty FBR assignments created by the optimizer there is a function:

#### Assignment General -> Reassign FBR

It will switch the position of all assigned trips that are in the wrong order, based on the rule. Only trips that are the same for both crew members will be switched.

# Max production days per vacation year

This chapter describes the functionality that limits the number of production days within a vacation year.

Note: This functionality applies to flight deck crew in variable group only.

## Background

Flight deck crew in variable group has a max limit of production days within a vacation year. In order to distribute the production days throughout the entire vacation year and to assure that production can be legally assigned in the last months, an algorithm based on a monthly target is used. The yearly limit is a hard limit and may not be exceeded. It is, however, calculated on a personal basis depending on availability, i.e. service grade, vacation, leave of absence etc.

## Rules

Two rules are used to limit the number of production days within a vacation year:

***ind\_max\_production\_days\_in\_va\_year\_fc\_all***

This rule limits the total number of production days in the entire vacation year. The limit is dependent on crew segment (Main or RC), set by the parameters:

Main: max\_production\_days\_in\_va\_year\_fd\_main\_p (default = 179)

RC: max\_production\_days\_in\_va\_year\_fd\_rc\_p (default = 185)

Note: The yearly limit is personally adjusted according to availability.

***ind\_max\_production\_days\_in\_month\_fc\_all***

This rule limits the number of production days in a calendar month based on the monthly target. The monthly target may be exceeded by a number of production days controlled by the parameter max\_overshoot\_production\_days\_in\_month\_p (default = 0). Assignments above the target are however penalized with a linear cost. The cost for each day is set by roster cost parameter 2.8 (Cost for each day exceeding the monthly limit of production days [FC]).

## Change rule calculation dates

The formulas calculating the yearly limit and the monthly targets are normally based on the start and end dates of the current vacation year. However, when crew changes group, segment or base within the current vacation year, the calculations for the max production days rule needs to start over from the date as the change occurs. This means that the rule will operate on a shorter period than a year and the calculations will be adjusted accordingly. New start and end dates for the rule are manually set in the table crew\_prod\_day\_change (see Table [4]).

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| crew | [reference] | Crew id, reference to crew |
| validfrom | Time | Start date for max production days calculations |
| validto | Time | End date for max production days calculations |
| currgroup | [reference] | Current group, reference to crew\_prod\_day\_groups |
| prevgroup | [reference] | Previous group, reference to crew\_prod\_day\_groups |
| si | String | Supplementary information |

1. crew\_prod\_day\_change

For the period between *validfrom* and *validto* these dates will override the default start and end dates given from the vacation year. If a new period is set, the dates used by the rule are calculated as: *start\_date = max(vacation\_year\_start\_date, validfrom)* and *end\_date = min(vacation\_year\_end\_date, validto).* The currgroup and prevgroup columns are only used to inform the planners of the reason to the new period. This means that the values set in these columns will not have an impact on the calculations.

Note: The calculations are performed on a monthly basis. If crew changes group, segment or base in the middle of a month, the first day of the following month should be used as change date.

**Example 1:**

Crew based in OSL (vacation year 01Jan-01Jan) changes segment from Main to RC at 01May2012. A new period is set in table crew\_prod\_day\_change according to:

|  |  |  |  |
| --- | --- | --- | --- |
| ***validfrom*** | ***validto*** | ***currgroup*** | ***prevgroup*** |
| *01May2012* | *01Jan2013* | *RC* | *Main* |

The calculations will now start over from 01May2012 as if this was the start of a new vacation year and the formulas will be adjusted to the new shorter eight months period. The new yearly target will be 185 \* (8 / 12) and the monthly targets are adjusted to reach this.

**Example 2:**

Crew based in CPH (vacation year 01May-01May) changes from FG to VG at 01Nov2012. A new period is set in table crew\_prod\_day\_change according to:

|  |  |  |  |
| --- | --- | --- | --- |
| ***validfrom*** | ***validto*** | ***currgroup*** | ***prevgroup*** |
| *01Nov2012* | *01May2013* | *VG* | *FG* |

**Example 3:**

Crew based in STO (vacation year 01Jun-01Jun) changes base from STO to OSL (vacation year 01Jan-01Jan) at 15Aug2012. A new period is set in table crew\_prod\_day\_change according to:

|  |  |  |  |
| --- | --- | --- | --- |
| ***validfrom*** | ***validto*** | ***currgroup*** | ***prevgroup*** |
| *01Sep2012* | *01Jan2013* | *OSL* | *STO* |

## Temporary service grade

In some cases it is necessary to override the normal service grade for a given crew member. This is typically used for part-time retirees that during a course work as fulltime crew. The temporary service grade is set in table crew\_prod\_day\_sg (see Table [5]).

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| Crew | [reference] | Crew id, reference to crew |
| validfrom | Time | Start of temporary period |
| validto | Time | End of temporary period |
| exclude\_days | Int | Number of non-service days within the period |

1. crew\_prod\_day\_sg

The temporary service grade within the period *validfrom* to *validto* will be calculated as:

*temp\_sg = (validto - validfrom - exclude\_days) / (validto - validfrom)*

Note: This temporary service grade only applies to the calculations max production days rule.

**Example:**

|  |  |  |
| --- | --- | --- |
| ***Validfrom*** | ***Validto*** | ***exclude\_days*** |
| *01Sep2012* | *01Oct2012* | *8* |

The temporary service grade in Sep2012 will be calculated as (30 - 8) / 30 = 0.73.

## Exclude crew

A crew member may be excluded from the max production days rule. This is done in table rave\_string\_param\_set by defining a period of exclusion for ravevar *p\_days\_skn\_exclude\_crew.*

**Example:**

To exclude crew 12345 in year 2012 the following entry should be added to table rave\_string\_param\_set:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Ravevar*** | ***val*** | ***validfrom*** | ***validto*** |
| *p\_days\_skn\_exclude\_crew* | *12345* | *01Jan2012* | *01Jan2013* |

## Report

The report ProductionDaysInfo is described in ‘Functional Reference –Reports’.