System Reference Manual

Jeppesen Crew Management System at SAS

Version 2.4, 2012-10-19

Table of contents

[About this document 1](#_Toc338419303)

[Overview 1](#_Toc338419304)

[Conventions 3](#_Toc338419305)

[System overview 5](#_Toc338419306)

[Work areas 6](#_Toc338419307)

[Client applications 7](#_Toc338419308)

[Server applications 10](#_Toc338419309)

[System internal data flow 12](#_Toc338419310)

[Customization layer – the CARMUSR 15](#_Toc338419311)

[CARMUSR directory structure 15](#_Toc338419312)

[Executable scripts 16](#_Toc338419313)

[Rave code 17](#_Toc338419314)

[Report code 17](#_Toc338419315)

[General python code 18](#_Toc338419316)

[Configuration 20](#_Toc338419317)

[Menu definitions for Studio 22](#_Toc338419318)

[Form definitions 22](#_Toc338419319)

[Database schema extensions 22](#_Toc338419320)

[Other files and directories 23](#_Toc338419321)

[Special customization 25](#_Toc338419322)

[Server environment configuration 25](#_Toc338419323)

[Jeppesen Resource System 27](#_Toc338419324)

[Jeppesen common configuration 28](#_Toc338419325)

[Menu definitions for Studio 31](#_Toc338419326)

[Leg Key matching 33](#_Toc338419327)

[Adding an airport to CMS 34](#_Toc338419328)

[System dependencies 39](#_Toc338419329)

[File based data in CMS 39](#_Toc338419330)

# About this document

This document describes the overall design of the Jeppesen Crew Management System at SAS.

#### Readers

The intended readers are:

* Developers
* System Administrators

#### Maintenance

This document is developed by Jeppesen Systems during an implementation project, but after a hand over the ownership is transferred to SAS.

The following person is responsible for the maintenance of this document:

Jonas Carlsson, Jeppesen

#### Version history

| Version | Release Date | Comment |
| --- | --- | --- |
| 2.0 | 2009-01-21 | Rewrote the entire document and changed layout to the Jeppesen standard layout. |
| 2.1 | 2009-06-09 | Added External Publish Server |
| 2.2 | 2009-12-08 | Various minor changes/updates |
| 2.3 | 2010-03-08 | Added information about how to add a new airport to the CMS system. |
| 2.4 | 2012-09-14 | Added changes due to Interbids 5 |
| 2.5 | 2012-10-15 | Initial update for CMS2. |

### Overview

This document is the main document for the Jeppesen Crew Management System at SAS and gives an overview of the full system. Detailed descriptions of applications, functionality and administration are available in the following documents:

* Functional Reference Manuals for the system:

Functional Reference Manual Common  
Detailed information about functionality shared between several applications. This includes applications for editing crew data, description of training processes as well as description of graphical markers in the shared applications.

Functional Reference Manual Integration  
Detailed information about functionality implemented for the integration part of the system. This includes applications for generating and maintaining salary and meal reports.

Functional Reference Manual Pairing  
Detailed information about functionality implemented for the pairing part of the system. This is a complement to the User Guide Pairing.

Functional Reference Manual Rostering  
Detailed information about functionality implemented for the rostering part of the system. This is a complement to the User Guide Rostering.

Functional Reference Manual Tracking  
Detailed information about functionality implemented for the tracking part of the system. This is a complement to the User Guide Tracking.

Functional Reference Manual Manpower  
Detailed information about functionality implemented for the Manpower application. This is a complement to the User Guide Manpower.

Functional Reference Manual PreStudio  
Detailed information about functionality implemented for the PreStudio application.

Functional Reference Non Core  
Detailed information about functionality implemented for the NonCore parts, including Salary and Meal applications.

* System Administrator Manual  
  Detailed information about how to setup, configure and maintain the system. This is a complement to the general System Administration Guide.
* Carmen Unified Data Model for SAS  
  Detailed information about how the data is stored in the database.

This document contains the following chapters:

* System overview; a high level walkthrough of the system and all components.
* Customization layer; a high level overview of the CARMUSR and the different parts of the customization layer.
* Special customization; a break down of the major customer specific customization.
* System dependencies; description and structure of depending parts.

### Conventions

This document uses the following conventions.

#### Tags

The following tags are used:

1. Matters of specific importance.
2. Cross-reference.

#### Type styles

The following type styles are used:

|  |  |
| --- | --- |
| Italic | cross-reference |
| Bold | Jeppesen components; such as commands, names of buttons, boxes, field names and menu items |
| Initial Capitals | names of Jeppesen windows and services |

# System overview

A Jeppesen Crew system consists of two main parts, the core and a customization layer. The core is the fundamental, common part of the system while the customization layer contains the customer specific configuration based on the core functionality.

The Jeppesen Crew Management System (CMS) consists of several different applications interacting. The main system parts are defined in the core while the configuration of how the interaction should be done is mainly defined in the customization layer.

**Client Applications**

Optimizers

(file based)

Alert Server

**Jeppesen Crew Management System at SAS**

**Other SAS systems (external)**

DIG

TIBCO

Scheduler

…

…

Crew Portal

1. An application overview of the Jeppesen Crew Management System as it is defined for SAS, divided by the client and server applications. The Sysmond controlled applications are specially marked.

Session Server

Dave

**Server Applications**

Retort Servers

Report Servers

Report Servers

Report Servers

Report Servers

Report Servers

Model Servers

Model Servers

Model Servers

Model Servers

Model Servers

Model Servers

Model Servers

Model Servers

Tracking Studio

Pre Studio

Table Editor

Planning Studio

Alert Monitor

Crew Info

Block Hours

Salary

Crew Meal

Crew Training

Man-power

Hand-over

Alert Generator

Recurrent updates

Sysmond

External Publish

Jeppesen

Crew Portal

Crew Portal DB

Looking at the applications, the system can be divided into client applications and server applications. The client applications are accessible by the user to view and edit data while the server applications are running in the background.

Typical client applications are Planning Studio, Crew Info and Alert Monitor. Typical server applications are Alert Server, Report Server and Model Server. A special client application is the SessionServer Launcher, where all other applications are started. A special server application is Sysmond, used for keeping a selection of the server applications up and running.

### Work areas

CMS is divided into several working areas with different responsibilities. The areas are dependent on the time interval one works in as well as the tasks one works with.

Follow up

Released rosters

Production planning

Long term planning

Pre Planning

Planning

Tracking

Post Planning

1. An overview of the different working areas and their time dependent relations.

JAN

FEB - MAR

APR

MAY …

The main planning areas (Pre Planning, Planning and Tracking) are time dependent. This means that those areas usually never work with the same assignment data. The Post Planning area administrates all reports generated for external systems as well as data related to those reports. This requires that they work in the whole period from roster release up until the final salary runs are done.

The system gives different access to the system dependent on what area one is working in. For each area there is a defined role giving access to different client applications. There is also an administrative role giving access to all applications.

Most roles have some special applications specially designed for the problems handled by the intended area. There are also more general applications available for most of the roles.

A list of all roles available in CMS:

| Role | Description |
| --- | --- |
| PrePlanner | The role used for long term planning. The main applications included in this role are Manpower and Pre Studio. |
| Planner | The role used for crew schedule planning ending in rosters released to crew. The main application in this role is Planning Studio. |
| Tracker | The role used for roster maintenance from rosters being released to crew to the actual day of operation. The main applications in this role are Alert Monitor and Tracking Studio. |
| PostPlanner | The role used for administrative work when it comes to ordering meal, hotels, etc. Also used for post processing before salary runs and similar. The main applications are Salary, Meal Order and Tracking Studio (with some added menu options). |
| Administrator | An administrative role given access to all applications. The Studio applications will in this role have more functionality then the applications when accessed from the normal roles. |
| CrewBaseAdm | A role giving access to some special, restricted tables in the database via the Table Editor application. |
| Viewer | A role giving view access to the Crew Info application. |

### Client applications

There are several client applications available in CMS. Some are standard applications and some are specially built to cover the need of SAS.

#### Launcher

Launcher is the entrance into all Jeppesen systems and it is the client part of the Jeppesen SessionServer. From the launcher the user gets access to a selection of applications, depending on the role defined in the system.

The launcher is an application running on a Microsoft Windows system.

#### Alert Monitor

Alert Monitor gives an overview of all alerts (illegalities and warnings) over a specified period. From the Alert Monitor a Tracking Studio can be opened to handle a selection of alerts.

This application is a standard application provided in the core of the system and it is a fundamental part of the Tracking system.

1. More information about the SAS specific configuration of the Alert Monitor is found in Functional Reference Manual Tracking.

#### Manpower

Manpower is the main application for long term resource planning, such as vacation and transition planning, with possibility to visualize the resources over a longer period.

It is a standard application provided as part of the core, although it is a Microsoft Windows application and needs to be installed on the same system as the Launcher is installed.

1. More information about Manpower is found in Functional Reference Manual Manpower.

#### Studio based applications

The main application for manipulating crew rosters is Jeppesen Studio and the core of that is provided as a standard application. The application can be configured differently when it comes to look, behaviour and available tools. In CMS for SAS Studio comes in three different versions: Planning, Pre and Tracking.

##### Planning Studio

Planning Studio is the application used for creating crew rosters before release to crew. The application provides access to crew and flight data as well as optimizers and many tools for manually manipulating the data. The same application is used for creating pairing and for creating rosters.

This application is a fundamental part of the Planning system.

1. More information about Planning Studio is found in Functional Reference Manual Pairing and Functional Reference Manual Rostering.

##### Pre Studio

Pre Studio is a Studio version with very limited functionality. The application is to be used for creating long term assignments on crew as well as updating activities far into the future. There is no possibility to work with production in Pre Studio.

1. More information about Pre Studio is found in Functional Reference Manual PreStudio.

##### Tracking Studio

Tracking Studio is the application used to maintain the crew rosters after they have been released to crew. The application provides tools for updating the data in a quick way where manipulations of crew assignments can be done down to single assignment objects.

This application is a fundamental part of the Tracking system.

1. More information about Tracking Studio is found in Functional Reference Manual Tracking.

#### Table Editor

Table Editor is an application for viewing and editing data in the database table by table. It can be invoked both standalone from the Launcher, as well as from within Studio.

This application is a standard application provided in the core of the system.

1. More information about Table Editor is found in Functional Reference Manual Common.

#### Wave based applications

Wave is a technology to allow for creating applications for data manipulation. All Wave applications available in CMS are specially built to cover the special need of SAS.

Many of the Wave applications are also available from inside the different Studio applications.

##### Crew Block Hours

Block Hours is an application visualizing the worked block hours each month for the different aircraft types for a selected crew.

1. More information about Crew Block Hours is found in Functional Reference Manual Common.

##### Crew Info

Crew Info is an application for visualizing and editing fundamental crew data, e.g. qualifications, contracts, employment, contact data.

1. More information about Crew Info is found in Functional Reference Manual Common.

##### Crew Meal

Crew Meal is an application for creating crew meal orders. The application allows for creating and managing the orders.

1. More information about Crew Meal is found in Functional Reference Manual Non Core.

##### Crew Training

Crew Training is an application for viewing and editing some parts of the training information for a selected crew.

1. More information about Crew Training is found in Functional Reference Manual Common.

##### Handover Report

Handover Report is an application for recording and retrieving information of importance at a shift change in the tracking operation.

1. More information about Handover Report is found in Functional Reference Manual Tracking.

##### Salary

Salary is an application for creating and managing salary reports, e.g. per diem and overtime. The application also allows for doing reruns of some of the reports.

1. More information about Salary is found in Functional Reference Manual Non Core.

#### Jeppesen Crew Portal

Jeppesen Crew Portal is the interface crew uses to register bids, preferences and requests that can affect planning.

Jeppesen Crew Portal is designed as several JavaEE web applications meaning that it needs a web application server to run. In the SAS deployment, the application server is JBoss.

Jeppesen Crew Portal is accessed from a link in the SAS Crew Portal.

#### Other client applications

There are some other client applications in the system possible to use and accessible by some roles.

##### Xterm

This is a Linux terminal window giving access to the Linux servers CMS is running on.

##### File reader

This is a way to access files created on by the system on the server side. It is a way to start the web browser Firefox on the client side with a path to the report area on the Linux servers.

1. More information about File reader is found in Functional Reference Manual Non Core.

### Server applications

There are several applications running at the server side, as background processes. They are for instance used for receiving data from external systems as well as providing the external systems with data. They are also used for keeping the fundamental parts of the system up and running and for updating the data internally.

#### Alert Generator

Alert Generator detects illegalities in the relevant period and makes them available for the Alert Monitor via the Alert Server.

The Alert Generator is a fundamental part of the Tracking system and needs to be running at all times thus it is controlled by Sysmond.

#### Alert Server

Alert Server is the server application for the Alert Monitor and handles the communication between the Alert Monitor client program and the Alert Generator but also between the Alert Monitor and the database.

The Alert Server is a fundamental part of the Tracking system and needs to be running at all times thus it is controlled by Sysmond.

#### Sysmond

Sysmond is a monitoring program used to make sure all system required server applications are up and running. If any controlled application stops, Sysmond will try to restart it.

Sysmond also starts recurrent update jobs for the data in the database as well as clean up jobs for the file system.

#### Data Integration Gateway

The Data Integration Gateway (DIG) is the framework for handling communication with systems outside of CMS.

DIG is a fundamental part of CMS and needs to be running at all times thus it is controlled by Sysmond.

#### Model Server

Model Server is the server application for all Wave based client applications as well as the Manpower application and the Table Editor. It is the application handling the actual communication with the database. The application is configurable and might behave somewhat differently depending on what application it is running.

Every client application started will start a separate instance of the Model Server.

#### Report Server

Report Server is used to produce data to external systems, although the distribution of the data is handled by DIG. A Report Server can load different states of the data as well as different time periods, depending of the need, and DIG will make sure the correct server is used.

In the SAS system several Report Servers are running with different data states as well as with different time periods.

The Report Servers are a fundamental part of the CMS system and needs to be running at all times thus it is controlled by Sysmond.

#### External Publish Server

The External Publish Server is a studio server continuously re-publishing crew affected by certain events reported by external interfaces. The purpose is to make sure that the latest flight and meal stop information will be visible in the Crew Portal, and to notify ad-hoc changes in check-in times. The re-publishing mechanism is designed as a client-server solution where clients (DIG channels) communicate with the server through a database table in which re-publication orders are issued.

#### Scheduler

Scheduler is the application used to start event triggered generation of external data. The Scheduler is configured to detect certain changes to the data in the database. When those changes occur the scheduler will order a report from a Report Server and then have DIG distribute it to the correct receiver.

The Scheduler is a fundamental part of the CMS system and needs to be running at all times and it is because of that controlled by Sysmond.

#### SessionServer

SessionServer is the application handling requests for accessing CMS. It can handle a multiple amount of systems.

The SesionServer is actually a separate system and a small customization area and is installed in a different place then the rest of the system. The customization part for the SessionServer system is just pointing out where the CARMUSR is located to enable the SessionServer to start the system.

1. More information about SessionServer is found in CMS System Administration Guide.

#### JBoss

JBoss is the JavaEE web application server where Jeppesen Crew Portal is deployed.

### System internal data flow

Data can either be viewed or edited. Usually the applications in CMS are used both for viewing and updating data in the system.

Viewing data can either be for making it possible to do updates or for sending data to external interfaces. But viewing data can also trigger a data change either by the request for the data or for system internal reasons.

Data updates can be triggered either via user updates through client applications, interfaces from external systems or internally triggered background applications. Depending on the type of change, it will be handled somewhat differently.

#### Client application data access

Almost all client applications available for the user can be used to access and update data. Usually a user application needs to view the data before any updates can be done, although the viewed data might not necessarily be the data that is updated. All client applications need to do a manual refresh to view the latest changes, if nothing else is mentioned.

Also almost all client applications available for the user can be used to do updates in the CMS database. Changes done in a client application are applied to the database when performing a save operation. The updates will directly be available to everyone using the system as long as the application used to view the data is refreshed.

The data is committed to the database either via a server application or directly to Dave. All Studio based applications commits changes directly to Dave whilst all other applications uses a server process to commit the changes.

#### Interface data access

SAS has several external systems in need of data from CMS. The out data can either be generated because of a request from an external system or by an internal trigger.

If CMS obtains a request message, DIG will pass it to an appropriate report server where a request will be generated and passed back to DIG for distributing it to TIBCO for further processing. The report server to choose depends on what type of data to handle as well as the time interval for the included data.

There is a lot of data needed by different systems that is delivered based on either events in the CMS data or recurrently. This is handled by the scheduler, telling the report server to generate reports.

All interfaces provide CMS with updated information through DIG. DIG gets the vast majority of data updates from message queues (MQ) but also files provided by external systems directly on CMS file system or via ftp are used to provide updated data. Usually updates do not require responses, but sometimes a request for data still generates a data change to provide the possibility to keep track of what has been delivered.

#### Internally triggered data access

Part of the data contained in CMS is data to be used for internal use only. This data is updated either when changes are saved in other applications, by recurrent reports generated by the scheduler or by recurrent data updates done by Sysmond.

# Customization layer – the CARMUSR

The Jeppesen Crew Management System (CMS) consists of two main parts, the core (CARMSYS) and a customization layer (CARMUSR). The core contains an identical base in all Jeppesen systems, although there are different versions of the core for tracking and for planning. The customization layer contains the customer specific configuration based on the core functionality. In this chapter the customization layer will be described from an overview level.

### CARMUSR directory structure

The CARMUSR of CMS for SAS can be divided into several areas: rave code, report code, python code for functionality, configuration, Jeppesen resources, menu definitions, select filters, executable scripts, form definitions, data schema extensions definition.

A listing of the most important directories and their content:

| Directory | Content |
| --- | --- |
| bin | Almost all executable scripts in the CARMUSR (see also etc/scripts). Both scripts intended for running from the command line and started by the system. |
| crc/etable | Configuration etables available to the system. The directories initial\_load, manpower and sas\_dump are only used when creating an initial database schema and are never used after that. The other etables are either configuration tables for Studio functionality or fall back tables if a table is missing. |
| crc/source crc/modules crc/report crc/require | All rave code for the different part of the system. Defines all legality but also lots of visualization in Studio and much of the data in reports. |
| crg | Report code for old report language. |
| lib/python/ report\_sources | Report code for the majority of the reports. |
| etc | Common configuration files for the CMS system. |
| etc/scripts | Scripts for starting the CMS infrastructure server processes under Sysmond control, such as reportserver, alertgenerator etc. These are mostly wrappers for CARMSYS scripts. |
| menu\_scripts | Studio menu definition files. |
| resources | Configurations or the Jeppesen Resource system. |
| lib/www/Manpower | Definition of data dialogs for Manpower. |
| lib/python | All python code defining functionality in the system. |
| lib/webapp | Files used to build the J2EE web applications used by the system. |
| lib/java | All java files used by the system. |
| data/form | Wave application layout definitions. New and redefined Studio forms (both CFH and Wave based). |
| data/manpower | Customized layout definitions for Manpower. |
| data/config | Different configuration files for the system. |
| data/config/ models | Extensions to the standard database schema definition. |
| data/config/ views | Redefinitions to the Table Editor user interface. |
| data/config/ Xresources | Icons and redefined X Resources for Studio. |

1. For more information about the different directories and types of code that are not described in this document, see Jeppesen System Configuration Reference Manual.

### Executable scripts

The executable scripts available in the CARMUSR are mainly wrapper scripts for functionality available in the CARMSYS, many of which are run either by an administrator or recurrently by Sysmond.

The administrative scripts are for maintaining the database schema, for updating the user configuration for the system, building rave rule sets from command line and for handling Sysmond, among other things. The recurrent task scripts are for nightly updates of the data, for cleanup of old data and for making data accessible.

Besides these scripts there are also some scripts used for starting applications like Model Server and Studio as well as the launcher and trip servers for InterBids.

1. More information about the administration scripts are found in the System Administrator Manual.

### Rave code

The user has nine different rule sets and one super rule set:

* BuildAcRotations for creating aircraft rotations
* Pairing\_FC for pairing, flight deck
* Pairing\_CC for pairing, cabin crew
* Rostering\_FC for rostering, flight deck
* Rostering\_CC for rostering, cabin crew
* Tracking\_FC for tracking and Report Servers, flight deck
* Tracking\_CC for tracking and Report Servers, cabin crew
* Manpower for manpower planning, both flight deck and cabin crew

The Tracking\_FC and Tracking\_CC are combined into a super rule set called Tracking. That rule set is the one used for Tracking Studio as well as most of the background processes requiring rule sets (e.g. Alert Generator, Report Servers and most of the recurrent updates).

Each rule set consists of a number of files, where each file contains a sequence of definitions. The files in a rule set must be complete in the sense that every identifier and variable mentioned is also defined in the code. You can use the Rave IDE to see all files included in the rule set.

All rule sets share some code, but the Manpower rule set has mainly separate code since the rave dialect differs between the Studio applications and the Manpower application.

### Report code

There are two different report languages in the Jeppesen systems, the Page Description Language (PDL) and the Python Report Toolkit (PRT).

PDL is the old report language used by the Jeppesen system and is being phased out. In CMS for SAS, there are a limited number of PDL reports and the vast majority of reports are done using PRT.

#### Page Description Language (PDL)

The report source code files for PDL are located in the crg directory in the CARMUSR, with the following sub directories:

| Sub-directory | Content |
| --- | --- |
| crew\_window\_general | Crew reports for all crew. Contains only bid and fairness reports and the planning report for rule exceptions. |
| crew\_window\_object | Crew reports for a single crew. Contains only the planning report for rule exceptions. |
| crr\_window\_general | Trip reports for all trips. Contains reports for statistics on trips and an analysis report for the TOR imitation package. |
| crr\_window\_object | Trip reports for a single trip. Contains reports for soft lock details and trip statistics. |
| hidden | Reports called from scripts. Also includes reports visualizing information in different windows in Studio. |
| include | Common parts shared between different reports. Usually this is where the main part of the report code is located. |

#### Python Report Toolkit (PRT)

The Python report source code files are located in the directory lib/python/report\_sources in the CARMUSR, with the sub directories:

| Sub-directory | Content |
| --- | --- |
| assignment\_window\_object | Activity reports for a assignment on crew. |
| crew\_window\_general | Crew reports for all crew. |
| crew\_window\_object | Crew reports for a single crew. |
| crr\_window\_general | Trip reports for all trips. |
| crr\_window\_object | Trip reports for a single trip. |
| report\_server | Reports called from the report server. |
| hidden | Reports called from scripts and from tracking Studio. |
| include | Common parts shared between different reports. |
| leg\_window\_general | Leg reports for all legs. |
| manpower | Reports used by the Manpower application. |

### General python code

The python files available in the CARMUSR are used for any type of automation, both within existing applications as well as separate processes. But python is also used for reports, as described above, and for editing some core functionality.

All of the python files in the CARMUSR are located in the lib/python directory. The majority of python code is located in different directories. Some files are located directly in the lib/python directory either for historical reasons or because of requirements from the CARMSYS.

The content of the lib/python directory is as follows:

| Directory/file | Decription |
| --- | --- |
| adhoc | Script/programs for handling various problems, mostly data related. |
| interbids | Files for the Jeppesen Crew Portal functionality. |
| carmdata | Functions for import CTF files and for filtering data in the TableEditor. |
| carmsdt | Different standard files included in all Jeppesen CARMUSR:s. |
| carmtest | Different tests for the CARMUSR functionality. |
| carmusr | The majority of the customized Studio functionality. |
| cio | Files related to check in, check out and editing of briefing times. |
| cmsadm | Files for manipulating systems user list. |
| crewlists | Files for generating crewlists. |
| CustomMenuStates.py | A file extending CARMSYS functionality for menu states. |
| dig | Files for customization of the DIG handlers. |
| hotel\_transport | Files for generating hotel and transport report calculation and maintenance. |
| ImportCommandsExt.py | Data file import options in planning. |
| meal | Files for meal order report calculation and maintenance. |
| MenuCommandsExt.py | Short commands accessed from menus in the different Studio applications. |
| passive | Files for passive booking report calculation and maintenance. |
| replication | Files for replicating data from CRU80 to CMS and vice versa. |
| salary | Files for salary report calculation and maintenance. |
| Select.py | Select functions used both in menus and in other functions. |
| StudioCustom.py | Defines python imports for Studio. It is an extension to the CARMSYS imports. |
| StudioServerCustom.py | Defines python imports for Studio servers. It is an extension to the CARMSYS imports. |
| tm.py | Customized wrapper functions for the CARMSYS table manager functionality.  Used by almost all functions accessing data within other applications. |
| TrackingGuiExt.py | Defines functions used in the CARMSYS to customize core functionality. |
| utils | Different helper functions. |

1. More information is found in each file.

### Configuration

The configuration for the system consists of different parts: it is configuration for the server environment and some different configuration systems for the clients and for interaction between the applications.

#### Server environment configuration

The main files for defining the server environment configuration to enable the system to start are:

| File | Descriptrion |
| --- | --- |
| CONFIG\_extension | CARMUSR specific extensions to the CARMSYS provided configuration. |
| CONFIG\_extension\_SAS | SAS specific extensions to the CONFIG\_extension. |
| etc/carmenv.sh | Defines what values several of the fundamental environment variables should have (e.g. CARMSYS, CARMTMP). |

Those files define what environment variables should be set on the linux system at start up as well as what Jeppesen specific environment variables to use.

1. More information about the server environment configuration in CMS for SAS can be found in the chapter on page 25.

#### Jeppesen resources

The Jeppesen resource system is mainly used by Studio to configure the behaviour of the system. In CMS for SAS, the resources can be applied for different roles as well as for the different products.

In the CARMUSR the Jeppesen resources are defined in the directories Resources/CarmResources and data/config/CarmResources.

1. More information about the use of Jeppesen resources in CMS for SAS can be found in the chapter on page 25.

#### Jeppesen common configuration

The Jeppesen common configuration is an XML-based system wide configuration. It can be used by all part of the system and the CARMUSR definition is an extension of an existing CARMSYS core definition.

The configuration is mainly used for server applications, but is also the only way possible to configure the Alert Monitor. The majority of the existing configuration is used for server configuration, such as DIG, the scheduler, Sysmond and the SessionServer.

In the CAMRUSR the common configuration files are located in the etc directory.

1. More information about the server environment configuration in CMS for SAS can be found in the chapter on page 28.

#### Manpower configuration

The manpower application uses the common configuration for information about what database schema to use. Changes to how the Manpower applications should behave is located in the file data/config/manpower.xml.

1. More information about the manpower configuration is found in the Functional Reference Manual Manpower.

#### Other configuration

In the CARMUSR directory data/config there are several other files used for configuration of applications or processes in the system.

| File | Description |
| --- | --- |
| area\_set | Defines all planning areas available in the planning system. Contains the same data as the database table area\_set.  This is a CMS SAS specific file. |
| crew\_categories | Defines the categories and positions available in Studio, and their relation. |
| CustomAttributes.xml | Defines the CARMUSR specific rave keywords for the Studio dialect of rave. |
| LegKeysAir | Defines what makes a leg unique when handled by Studio and several server applications. |
| XResources/StandardCas XResources/StandardCct | Defines extensions to the X resources used by the planning, pre and tracking Studio.  The StandardCas is used by Planning and Pre while the StandardCct is used by Tracking. |
| SGEOptions.etab | Defines available optimization queues available for optimization in the planning phase. |

1. More information about these configuration files can be found in the *System Configuration Manual*.

### Menu definitions for Studio

The menus in Studio are dependent on the application started as well as the role of the user starting the application. There are menus defined for the applications Planning, Pre, Tracking and Server, where the Server definition is only available for Studio based background processes.

In additions there are additional menus added to the applications for the roles Administrator and PostPlanner.

1. More information about the menu definitions for Studio in CMS for SAS can be found in the chapter on page 31.

### Form definitions

In the CARMUSR there are several redefinitions of forms available for the user. Those redefinitions are located in the following directories:

| Directory | Description |
| --- | --- |
| lib/www/manpower | Contains definitions of Wave forms used internally by Manpower. |
| data/manpower | Contains definitions of information in the dialogs for Manpower. |
| data/form | Contains redefinitions of forms available in Studio as well as the Wave based applications started from the launcher. |

### Database schema extensions

The database schema structure defined in the CARMSYS is extended with additions needed by CMS for SAS. The extensions include both additions to existing tables as well as entirely new tables. The extensions are located in the directory data/config/models.

1. More information about the database extensions is found in the document Carmen Unified Data Model for SAS.

### Other files and directories

In the CARMUSR there are several files containing minor, special settings usually available in a normal Jeppesen system.

| File/Directory | Description |
| --- | --- |
| data/Airport | Directory containing definitions of all airports available in the system. The data is also contained in the database and the database is master of the data. |
| matador\_scripts | Directory containing scripts to be used by the rostering optimizer. |
| select\_filters | Directory with different select filters available from inside Studio. |
| Testing | Configuration for the test tool texttext, used for simple automatic tests of the CARMUSR. |
| data/config/ LegKeysAir | A file used for defining how to match assigned legs with actual activities. |

1. More information about the LegKeysAir file is found in chapter on page 33.

# Special customization

The Jeppesen Crew Management System for SAS (CMS) has some special customization to handle problems not covered by the core functionality. Much of this is handled by python scripts, but some parts are done by setting up different configurations in a special way. Those special configuration settings will be described a bit more in this chapter.

### Server environment configuration

The server configuration is mainly defined in the extension to the CARMSYS configuration (CONFIG\_extension and CONFIG\_extension\_SAS) and the configuration file etc/carmenv.sh.

In these files there is configuration determining what CARMSYS to use as well as what menus to use, among other things.

#### Environment variables

There are some core environment variables needed by a Jeppesen system to run. There are also some environment variables specially defined for CMS for SAS. The following environment variables are the most important once used in CMS for SAS:

| Environment variable | Description |
| --- | --- |
| CARMUSR | The path to the CARMUSR to be used. |
| CARMSYS | Path to the CARMSYS to be used. |
| CARMTMP | Path to the temporary area where rule sets, logfiles, etc are stored. |
| CARMDATA | Path to the directory where filebased data is stored. This data is non-static data used to create file plans or to import information about crew. It also contains information about data plans. |
| SK\_APP | The application to start. This is a SAS specific variable with the possible values Tracking, PrePlanning, Planning, Manpower and Server. |
| PRODUCT | The variable defining what behaviour the core Studio should have. In CMS for SAS it is set depending on the SK\_APP.  In the SAS system, the value can be one of Cct, Cas and Pac. |
| CARMROLE | The role a specific user is assigned. If started from the launcher, the role is assigned the value of the SessionServer created environment variable CCROLE. |
| CARMCCSITE | A SessionServer defined variable defining what site the system is running at. At SAS this value should always be SAS.  The variable SITE contains the same information. |
| CARMCCSUBSITE | A variable used to define what sub site at SAS the system is running at. Can have the values PROD, LIVETEST, TEST and DLYTEST. |

If the file etc/carmenv.sh is sourced, the only environment variable needed is CARMUSR variable to get all the other variables defined. But the CARMSYS and CARMTMP variables are depending on what values the SK\_APP variable is set to.

#### CARMSYS usage

The SAS specific configuration requires the usage of several different CARMSYS. This is to enable changing a CARMSYS for one set of functionality without changing it for the full system.

In the CARMUSR there needs to be several links to CARMSYS:

| Link name | Description |
| --- | --- |
| current\_carmsys\_cct | Link pointing to the CARMSYS to be used for the tracking client applications and all wave applications started from the launcher.  Used for Tracking Studio, AlertMonitor and all Wave based client applications. It is also used for the model servers used by the Wave based applications. Requires a current\_carmtmp\_cct link. |
| current\_carmsys\_cas | Link pointing to the CARMSYS to be used for Planning Studio as well as the optimizers. Used for Planning Studio and the optimizers.  Requires a current\_carmtmp\_cas link. |
| current\_carmsys\_cmp | Link pointing to the CARMSYS to be used for Manpower.  Used for the Manpower model server and some recurrent updates. Requires a current\_carmtmp\_cmp link. |

The tracking, planning and manpower links are required for the system to run, although the planning and manpower links can point to the same CARMSYS. If no integration link is created, the tracking link will be used instead.

To make it possible to have different CARMSYS for different parts of the system, there is also a need for different temporary areas (CARMTMP) for each CARMSYS. The temporary area is used for log files as well as compiled Rave rule sets. To support different CARMSYS there is a need to enable different rave versions making special temporary areas needed.

What CARMSYS to use is defined by the environment variable SK\_APP.

| SK\_APP value | Result |
| --- | --- |
| Planning PrePlanning | CARMSYS will be set to current\_carmsys\_cas and CARMTMP will be set to current\_carmtmp\_cas. PRODUCT will be set to Cas. |
| Manpower | CARMSYS will be set to current\_carmsys\_cmp and CARMTMP will be set to current\_carmtmp\_cmp. PRODUCT will be set to Pac. |
| Tracking | CARMSYS will be set to current\_carmsys\_cct and CARMTMP will be set to current\_carmtmp\_cct. PRODUCT will be set to Cct. |
| Server | CARMSYS will be set to current\_carmsys\_int and CARMTMP will be set to current\_carmtmp\_int. If current\_carmsys\_int does not exist, the setting will be the same as for Tracking. PRODUCT will be set to Cct. |

### Jeppesen Resource System

The Jeppesen resource system is used by several parts of the CMS system. The basis is defined in the CARMSYS core with extensions in the CARMUSR. The CARMUSR contains resources in three different directories:

| Directory | Description |
| --- | --- |
| data/config/CarmResources | Contains the Implementation.etab file with resources applied to all applications and which should never be changed. |
| Resources/CarmResources | Contains the main part of the resources for the CARMUSR customization. |
| CARMDATA/Preferences | The user specific resources. The resource file is named the same as the unix user account. . |

For CMS for SAS the resource part of the CAMRUSR have been extended to enable having resources modified depending on both application used as well as the user role. To enable this extension the CARMUSR part of the resource system is divided into several files:

| File | Description |
| --- | --- |
| ModeCrewBaseAdm.etab ModePostPlanner.etab ModePrePlanner.etab ModeTracker.etab | These files define the resources applied for the role a user is assigned.  The files for the roles Administrator and Planner is located in the CARMSYS. |
| Customer.etab | Resources applied for all applications and roles. |
| Manpower.etab Planning.etab Tracking.etab PrePlanning.etab Server.etab | These files define the resources applied for the applications. |

The resource files are read in the order listed above, with the Imeplementation.etab file first followed by the role specific resources, the Customer.etab resources, the applications specific resources and finally the preferences.

There is also a special resource file in the CARMUSR called UserDefault.etab. It is not read by the system but copied to the CARMDATA/Preferences directory the first time a users accesses the system or if no preferences file can be found.

### Jeppesen common configuration

The Jeppesen common configuration is an XML based configuration system used by most part of the system. In CMS for SAS the configuration is used to define settings for the SessionServer and Launcher, all server processes as well as part of the setting for most applications.

The configuration is located in the etc directory in the CARMUSR and can be divided in SessionServer configuration and application configuration.

Some of the files are used by all parts:

| File | Description |
| --- | --- |
| config.xml | The base file for the configuration in the CARMUSR. This is the file included in the CARMSYS configuration and needs to import all other files used in the configuration if not explicitly commented. |
| site.xml  site\_[some site].xml | These files contain site specific configuration and is where most of the initial configuration is done when starting up a system.  The site.xml file is only used to select the correct site file. It starts by looking for site and subsite. If no such file is defined it looks for a site file. |
| users.xml  users\_[some site].xml | These files define the users in the system and the roles they have.  The users.xml is only used to select the correct site dependent file. |
| roles.xml | The file defines what role a user has when logging into the system or starting an application.  Note: The order in which the roles are defined for a user will have impact on the configuration used when running an application. The first role found in the list, that has the rights to run the application, will determine the configuration.  Example: Both Pre- and PostPlanners have the rights start Planning Studio. If the user role is configured as in [1] Planning Studio will use the PrePlanner configuration. For [2] the PostPlanner configuration will be used.  [1]:  <user name="x" fullname="x">  <role>PrePlanner</role>  <role>PostPlanner</role>  </user>  [2]:  <user name="x" fullname="x">  <role>PostPlanner</role>  <role>PrePlanner</role>  </user> |

#### SessionServer configuration

The base of the configuration for the SessionServer is defined in the CARMSYS. In the CARMUSR there are extensions to the default configuration to make the SessionServer behave as expected for SAS.

The files used for configuring the SessionServer are:

| File | Description |
| --- | --- |
| server\_factory.xml | This file contains extended configuration to the server factory settings defined in the CARMSYS.  It contains information about how applications should be handled on the server side. |
| applications.xml  application\_alertmonitor.xml  application\_alertmonitorstudio.xml  application\_blockhours.xml  application\_crewinfo.xml  application\_crewmeal.xml  application\_crewtraining.xml  application\_daveexplorer.xml  application\_handover.xml  application\_manpower2.xml  application\_manpower.xml  application\_manpower\_admin.xml  application\_planningstudio.xml  application\_postplanner\_filereader.xml  application\_prestudio.xml  application\_salary.xml  application\_tableeditor\_admin.xml  application\_tableeditor.xml  application\_trackingstudio.xml  application\_xterm.xml | The applications.xml file is the main file included in the config.xml file and includes all application specific files. The application specific configuration files defines configuration relevant for the SessionServer to handle each application.  Different applications have more or less configuration in these files depending on the sort of application and the level of configuration. |

#### Application configuration

Each application running on the server side has its own configuration.

The files used for configuring the applications are:

| File | Description |
| --- | --- |
| programs/alertgenerator.xml | The main configuration of the alertgenerator. |
| programs/alertrpcserver.xml | The configuration for handling the alertserver (formerly known as the alertrpcserver). |
| programs/dave\_cleanup.xml  programs/dave\_truncate\_history.xml | The configuration for various Dave data cleanup scripts run at regular intervals. |
| programs/programs/  programs/jmp |  |
| programs/ |  |
| programs/dig.xml | The main configuration for DIG. |
| programs/ |  |
| sysmond/\* | The configuration for the different Sysmond instances and how they should behave. There is one file per target environment. |
| sysmond/tasks.xml  sysmond/crontab.xml | The recurrent tasks handled by Sysmond. |
| dig/channels/\* | The configuration the channels DIG works with. One file for each channel. |
| dig/digscheduler.xml | The configuration for recurrent tasks handled by the scheduler. |
| hosts/\* | The configuration for where Sysmond should run as well as what each Sysmond instance should handle. One file per target environment.  The main configuration (config.xml) will start by looking for a site and subsite file. If no such file is defined it looks for a site file. |
|  |  |
|  |  |
|  |  |

### Menu definitions for Studio

Studio has a possibility to customize the menus. In CMS for SAS this is used to make Studio behave differently depending on what Studio application it is being used as, as well as depending on the role the user has.

The Jeppesen Studio menu system usually consists of a CARMSYS located base file with a possibility to add in additions in the CARMUSR. The base file differs a bit depending on what value the environment variable PRODUCT is set to.

In the CARMSYS there are a lot of menus defined with different purposes. The menus in CMS for SAS are divided into core menus and other menus. The core menus are the menus defined by the applications and being accessible from Studio directly while the sub menus are only accessible from another menu.

In CMS for SAS the base file is moved to the CARMUSR. All menus are based on the standard menus provided in the CARMSYS, but each application has special modifications created specially for CMS for SAS. For each definition, there is a file defining what should be removed from the standard menus, what should be modified and what should be added.

For some of the roles there is also add-on menu options defined. The add-on does not remove anything from the menus but adds menu options and might also redefine them.

#### Menu file structure

The files for the menu structure are located in the menu\_script/menus directory. In the directory there are two directories used for this: application and customization.

##### application directory

Each application defined by the environment variable SK\_APP has a special menu definition file. The file includes all other files making up the menu definition for the specified application. This includes the CARMSYS core menus as well as the CARMUSR defined modifications.

The inclusion of the CARMUSR edits are done by including all relevant menu files from the customization directory.

The base name of the file must be the same as the environment variable SK\_APP and the suffix must be “.menu”, e.g. “Planning.menu”.

##### customization directory

Each application if CMS for SAS has a special set of menus. Those menus contain modifications and additions to the standard menus because of the special need defined for SAS. The customization directory contains those modifications. In the directory there is also files making it possible to add modifications based on the role the user is logged in as.

In the directory there is one file for each CARMSYS defined core menu modified in the CARMUSR. It is the main menu modification file and it is called the same as the menu (with the suffix “.menu”). The main file contains an import of the actual modification files, using the environment variable SK\_APP. The modification files are located in the directory application and have the same name as the main file but with the application name as a suffix, e.g. if the main file is TOOL\_BAR.menu the modification files would be TOOL\_BAR\_Planning.menu, TOOL\_BAR\_Tracking.menu and TOOL\_BAR\_PrePlanning.menu. In the main file the add-ons for the roles is also included by using the environment variable CARMROLE. The role files are in the directory role.

In the main file there should also be the possibility to add modifications common for all applications. This is done by also including a file with common definitions.

1. /\*   
   \* This is the main customization file for the core menu TOOL\_BAR.  
   \* The file is named TOOL\_BAR.menu.  
   \*/   
     
   /\* Common changes to be applied for all applications \*/  
   include - "$CARMUSR/menu\_scripts/menus/customization/common/TOOL\_BAR\_Common.menu"  
     
   /\* This is the inclusion of the application specific menu additions. \*/  
   include - "$CARMUSR/menu\_scripts/menus/customization/application/TOOL\_BAR\_$SK\_APP.menu"  
     
   /\* This is the inclusion of the role specific menu additions. \*/  
   include - "$CARMUSR/menu\_scripts/menus/customization/role/TOOL\_BAR\_$CARMROLE.menu"

In each of the modification files there are defined modifications of the core menu. Each modification file can also include menu specific sub menus, if the submenus are only used in the modified core menu. But if the submenu is used by several core menus it should be defined and used in the same way as the core menus.

1. The sub menu SelectCrew is a large menu used from both the assignment menu and the crew menu. It also differs between the different applications. Because if this is should be treated in the same way as a core menu and not as a sub menu.

### Leg Key matching

For many applications there is core functionality for matching assignments on crew to actual activities. This is called Leg matching and in the CARMUSR there is a possibility to customize how this should be matched. The matching is done by two different keys, a primary and a secondary key, where a match is tried for the primary key first and then for the secondary key. To enable this there is a possibility to define one set of keys for matching flight legs and another set of keys to match ground duties.

For CMS for SAS the leg keys are defined as follows:

1. Primary key for flight leg and ground duty and secondary key for ground duty:

The type of leg

The carrier

The flight number

The flight suffix

The start date of the leg

The departure airport

The departure time

The arrival time

The block time

1. Secondary key for flight leg:

The type of leg

The carrier

The flight number

The flight suffix

The start date of the leg

The departure airport

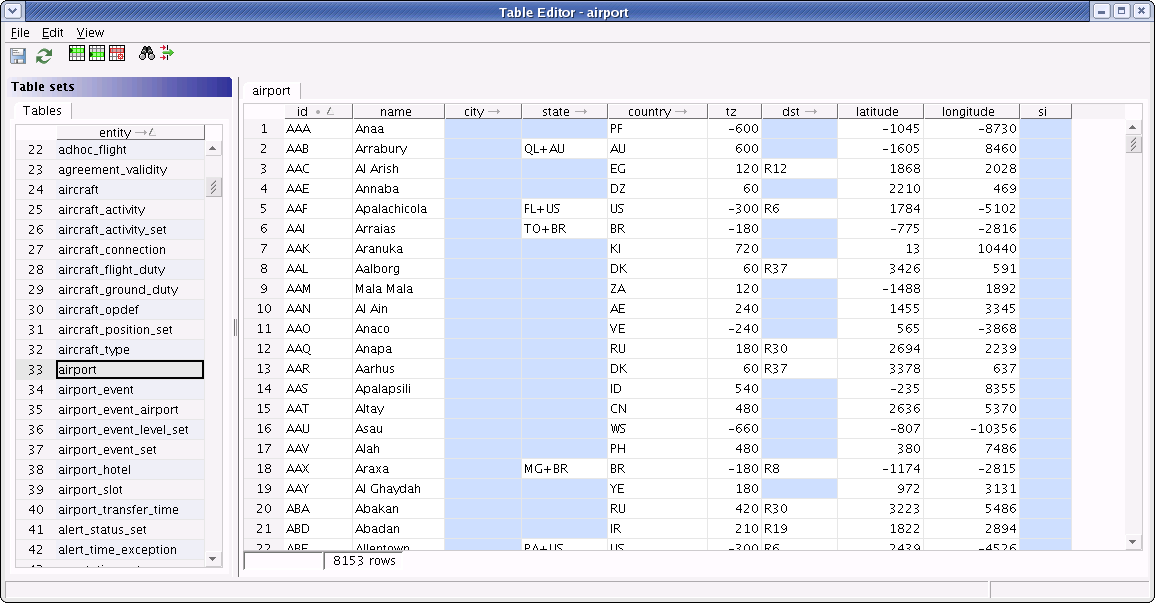
The definition of the keys and the defined order to be used is done in the file data/config/LegKeysAir in the CARMUSR and for the SAS system.

### Adding an airport to CMS

To add a new airport to CMS, it is necessary to add it both in the AirportManager and in the database table airport.

To add an airport in the AirportManager, see Carmen\_user\_guide – Airport Manager. Remember to compile and publish the airport file after adding an airport.

To add an airport to the database table airport, Open TableEditor, and choose airport table:



Add a new row, and update the id with the IATA airportcode, and fill in all other relevant information. The tz filed is the normal timezone difference from UTC, in minutes (+/-). The dst field is a reference to a rule in the dst\_rule table, choose an applicable dst from the drop down menu. Latitude and longitude are saved in arc minutes. To convert from degree/minutes to arc minutes, use the formula below:

Latitude = degree \* 60 + min, if latitude is S, multiply by -1

Longitude = degree \* 60 + min, if longitude is W, multiply by -1

# System dependencies

To be able to run the Jeppesen Crew Management System for SAS, there are some special requirements on the system except the normal requirement for a Jeppesen system. There is also a difference between the data storage of a system using a database compared to a classical file based planning system.

### File based data in CMS

The vast majority of the data in the CMS system is stored in the database. But some data is contained in files. Those files are either located in the crc/etable directory in the CARMUSR or in a special area called CARMDATA and are usually in a special format for external tables (Etables).

The files defined in the CARMUSR contain mainly static data and can mostly be seen as configuration data for the system. The data stored in CARMDATA is data related to crew or flights and might be updated at any time.

#### CARMUSR contained data

The external tables are used for data needed by Rave. Some of this data can be seen as source code because the close relation to the Rave code. These files should only be changed by developers and expert users and not planners. Because of this, all of these tables are located within the CARMUSR and as such are controlled by CVS.

The data is located in:

* $CARMUSR/crc/etable/.BaseConstraints  
  Base constraints default tables
* $CARMUSR/crc/etable/.BaseDefinitions  
  Base definitions default tables
* $CARMUSR/crc/etable/  
  Configuration tables for different parts of the system.

#### CARMDATA directories

CARMDATA mainly contains data related to crew, flights or similar data that is not part of configuration. But in CARMDATA there is also some other type of data, like preferences for the users and data exports. CARMDATA contains several directories and contains as described here:

| Directory | Description |
| --- | --- |
| app\_cache |  |
| CREW\_PLAN | Crew tables generated for exported planning plans. These are only used by file plans exported for optimization runs. |
| crew\_portal | Contains files exported and imported between Jeppesen Crew Portal and the rest of CMS. |
| crg\_saved |  |
| ETABLES | Tables with data from external systems. The data is mainly used for pairing production estimates but also data from InterBids for rostering is contained here. |
| FILE\_PLAN\_DATA |  |
| FP\_FILES | File based time tables. This type of data is usually stored in the database, but it is needed here to be able to work with file based long term planning and prognoses. |
| GROUND\_DUTY\_FILES | Tables for defining properties for ground-duty activities. These can be imported into the Rostering system. |
|  |  |
| LOCAL\_PLAN | All saved plans for planning as well as pointers to the database plan. Also includes scenarios for tracking.  File based optimization plans contain all relevant data from the database and the tables are mainly stored on local plan level. |
| Manpower | Contains all data Manpower data to and from InterBids. |
| MIGRATION\_DATA | Data migrated from SAS systems to the CMS system. This data will not used after production. |
| PLAN\_EXPORT | Contains data files for exports of the database. |
| preferences | Individual settings of colours, printers etc. |
| profiling | Profiling data. |
| REPORTS | Contains all reports generated by the system. |