

**Functional Reference**

*Document Type: Functional Reference*

*Version: 3.8*

*Last Changed: 24Feb 2014*

Table of Contents

[1 Specifications 5](#_Toc381010945)

[2 Definitions 6](#_Toc381010946)

[2.1 Run Type 6](#_Toc381010947)

[2.2 Salary System 6](#_Toc381010948)

[2.3 Administrative Code 6](#_Toc381010949)

[2.4 Run ID 6](#_Toc381010950)

[2.5 Job ID 6](#_Toc381010951)

[2.6 Export File 7](#_Toc381010952)

[2.7 Statement/reports 7](#_Toc381010953)

[2.8 Release File (Salary File) 7](#_Toc381010954)

[2.9 Salary Validity Check Report 7](#_Toc381010955)

[2.10 Balancing Report 7](#_Toc381010956)

[3 System Overview 8](#_Toc381010957)

[3.1 Batch Jobs and Manual Jobs 8](#_Toc381010958)

[3.2 Workflow 8](#_Toc381010959)

[3.3 Preparation 8](#_Toc381010960)

[3.3.1 Validity Check report 9](#_Toc381010961)

[3.3.2 Test runs 9](#_Toc381010962)

[3.4 Execution 9](#_Toc381010963)

[3.4.1 Admin Codes 9](#_Toc381010964)

[3.4.2 Salary Run 11](#_Toc381010965)

[3.4.3 Retro 11](#_Toc381010966)

[3.4.4 Cancel 12](#_Toc381010967)

[3.5 Validation 12](#_Toc381010968)

[3.5.1 Balancing Report 12](#_Toc381010969)

[3.5.2 Statements and Reports 12](#_Toc381010970)

[3.5.3 Salary Export 12](#_Toc381010971)

[3.6 Approval 13](#_Toc381010972)

[3.6.1 Release 13](#_Toc381010973)

[3.6.2 Auto-release 14](#_Toc381010974)

[4 Salary Application 15](#_Toc381010975)

[4.1 Prerequisites 15](#_Toc381010976)

[4.2 Start Salary Application 15](#_Toc381010977)

[4.2.1 File menu 16](#_Toc381010978)

[4.2.2 Reporting menu 16](#_Toc381010979)

[4.2.3 Actions menu 16](#_Toc381010980)

[4.2.4 Admin menu 17](#_Toc381010981)

[4.3 Start Salary Run 17](#_Toc381010982)

[4.4 Show Statement 17](#_Toc381010983)

[4.5 Cancel Run 17](#_Toc381010984)

[4.6 Retro Run 17](#_Toc381010985)

[4.7 Release Run 18](#_Toc381010986)

[4.8 Send e-mails 18](#_Toc381010987)

[4.9 Note about running on old periods 18](#_Toc381010988)

[4.10 Create an Export File 18](#_Toc381010989)

[4.11 Job viewer 19](#_Toc381010990)

[4.12 Reading Export Files 19](#_Toc381010991)

[5 Database layout 21](#_Toc381010992)

[5.1 Configuration tables 21](#_Toc381010993)

[5.1.1 salary\_admin\_code 21](#_Toc381010994)

[5.1.2 salary\_article 21](#_Toc381010995)

[5.1.3 salary\_region 22](#_Toc381010996)

[5.1.4 Salary Crew Exceptions 23](#_Toc381010997)

[5.1.5 Crew with Convertible Overtime 24](#_Toc381010998)

[5.2 Data Tables 24](#_Toc381010999)

[5.2.1 salary\_run\_id 24](#_Toc381011000)

[5.2.2 salary\_basic\_data 24](#_Toc381011001)

[5.2.3 salary\_extra\_data 24](#_Toc381011002)

[6 Recommendations 25](#_Toc381011003)

[6.1 When to use Retro 25](#_Toc381011004)

[6.2 When to use Cancel 25](#_Toc381011005)

Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Description |
| 1.0 | F. Acosta | 2006-03-21 | First version. |
| 1.1 | F. Acosta | 2006-03-28 | Updated after proof-reading. Parameters and resources are described more in detail. |
| 1.2 | F. Acosta | 2006-03-31 | Updated after review. |
| 2.0 | F. Acosta | 2006-05-30 | Added sections about overtime. |
| 2.1 | L. Wentzel | 2006-10-06 | Updated to database solution. |
| 3.0 | F. Acosta | 2008-08-20 | Document completely rewritten. |
| 3.1 | R. Tropp Larsen | 2008-10-07 | Added additional info for running on old periods. |
| 3.2 | K. Altsjö | 2008-12-09 | Simplified running on old periods |
| 3.3 | H. Vazquez | 2009-03-19 | Added release functionality |
| 3.4 | H. Vazquez | 2009-04-22 | Added Temporary crew run (CR 186) |
| 3.5 | F. Acosta | 2010-05-12 | Updated for CR 464. |
| 3.6 | S. Olsson | 2013-02-27 | Minor update for CMS2. |
| 3.7 | C-H Coakley | 2013-11-05 | General revision and clean up |
| 3.8 | C-H Coakley | 2014-02-24 | Revision § 6 |

# Specifications

The salary module combines several of the interfaces between CMS and the different salary systems. More details can be found in the specifications for each of these interfaces.

|  |  |
| --- | --- |
| **Interface number** | **Interface name** |
| 43.2 | AMBI list |
| 43.5.1.1 | Vacation days performed |
| 43.5.1.2 | Vacation days remaining |
| 43.5.2 | Vacation days (SE) |
| 43.5.3 | Vacation days (DK) |
| 43.5.4 | Vacation days (NO) |
| 44.1 | Perdiem |
| 44.2 | Inst. Allowance |
| 44.3 | Overtime and Allowances |
| 44.5 | Bought days and compensation days |

# Definitions

## Run Type

Each run has a *Run Type* that identifies the type.

* AMBI
* Comp. days
* Instructor’s allowance
* Overtime
* Temporary Crew Allowance
* Per Diem
* Vacation (Yearly) (cannot run manually)
* Vacation (Performed)
* Vacation (Remaining)

## Salary System

These are the available *Salary Systems*:

* DK – Denmark
* NO – Norway
* SE – Sweden
* JP – Japan[[1]](#footnote-1)
* CN – China1

## Administrative Code

|  |  |  |
| --- | --- | --- |
| N | Normal run | A run started with the intention that the export file will be sent to an external salary system. |
| T | Test run | A run started for testing or control purposes. |
| R | Retro run | See Retro below. |
| C | Cancelled run | See Cancel below. |

## Run ID

Each Run is identified by a Run ID. This is a sequential number that is handled internally. It is not possible to tell in advance what next Run ID will be.

## Job ID

Each job that is submitted to the job queue will get a unique job ID. The job id is not linked to the Run ID in any way. The job ID will show up in the user interface after the user has submitted a job.

## Export File

|  |  |  |
| --- | --- | --- |
| Flat |  | File format suitable for transfer to main frame computer. |
| CSV | .csv | Comma Separated Values, a format that can be read by e.g. Microsoft Excel |
| HTML | .html | File format suitable for Web browsers. |

The *Flat* file format will create a text file in a format that is defined by the salary system. The current implementation supports two different *Flat* file formats, one for Denmark and Norway (SAP format), and another one for Sweden (PALS format).

Export files can be recreated at any time. They will have a unique file name to avoid that old data will be overwritten.

## Statement/reports

A statement or report containing details about the run is created as part of a *Per Diem,*  *Overtime or TEMP Crew*  run. These statements are created in connection to a run to avoid differences between data sent to the salary system and the statement report. The statements are saved as PDF files on the server. For Swedish TEMP Crew an additional report is created, “Resource Pool Illness Report.

## Release File (Salary File)

A Release File is a file in *Flat* file format, which has been exported from a run id. The file is placed in a folder from where other systems can pick it up.

## Salary Validity Check Report

The Salary Validity Check Report is created every time a run is started. This report contains problems that might disturb the results. This report also contains general information about the environment.

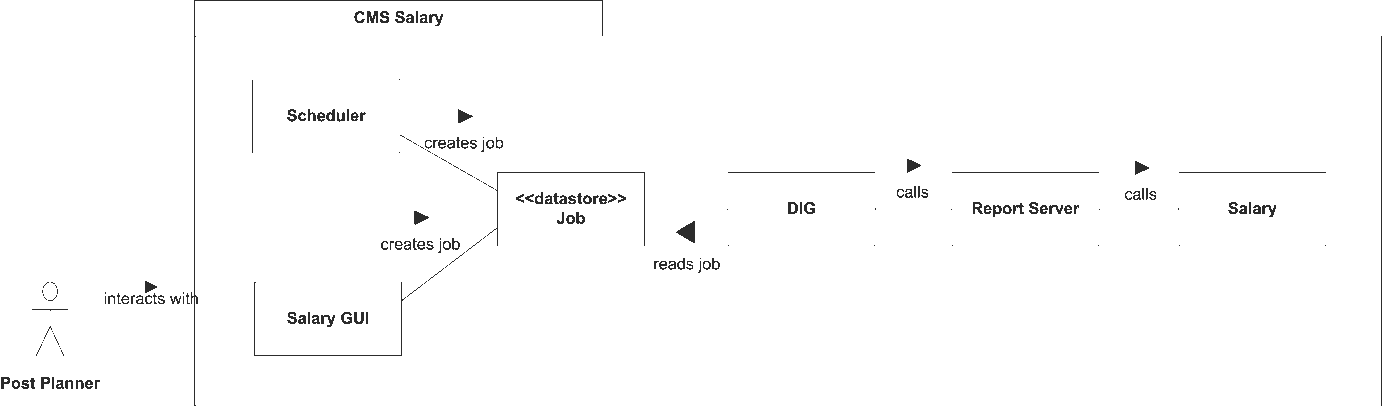
## Balancing Report

A summary email that is sent to a number of predefined addresses (*afstemningsunderlag*). This message is prepared and sent as part of the release procedure. The report can also be created interactively from the Salary GUI.

# System Overview

## Batch Jobs and Manual Jobs

The Salary system consists of several components show in the figure below.



Figure

The end user interacts with the Salary GUI. From the GUI it is possible to start jobs manually.

The Scheduler component adds batch jobs using a schedule that is described in the configuration file digscheduler.xml

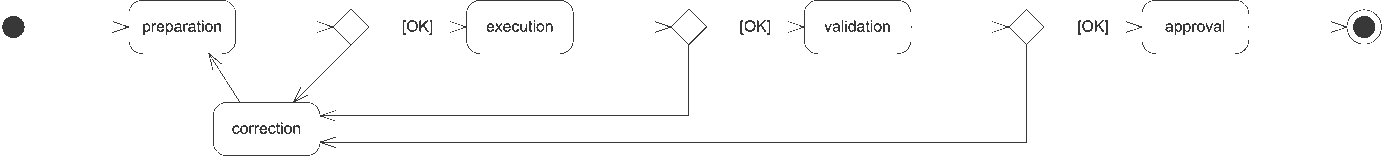
The DIG component reads the job queue at fixed intervals and invokes a Report Server instance to do the work.

The Report Server runs the Salary application.

## Workflow

Conceptually we divide the workflow into four major steps: Preparation, Execution, Validation and Approval. Each of these steps can contain sub-tasks.

The idea is that each step should carry the process forward, and allow for check-points.



Figure

If faults are found at a check-point, the errors have to be corrected and the process has to be restarted.

The final approval step makes the run “published”, data is propagated to external systems and (in some cases) accounts will be increased.

## Preparation

The purpose of the preparation step is to find faults that will spoil the result of a subsequent Salary run.

It is of no use to start a salary run (calculation) before we know that all involved crew have valid rosters, and that their assignments are correct.

### Validity Check report

The Salary Validity Check report is created as part of any run creation (see below). This report can also be created interactively from within the Studio application. This report contains some known problem sources that could interfere with a Salary run.

Current tests include:

* overlapping activities,
* full day activities that don’t start or end on a whole day,
* missing stations,
* off-duty / home base mismatch,
* extremely long trips,
* missing data in salary tables (e.g. salary codes (intartid/extartid and exchange rates),
* ground transportation where default value 1 hour is used
* and missing contracts.

### Test runs

Test runs (T) can be started at any time and they don’t disturb production. Reports are produced when applicable (Per Diem / Overtime / Temp Crew). These reports can be used to check validity.

Results from these Test runs can be exported to a generic format (Comma Separated Values, CSV). CSV files can be imported into e.g. Microsoft Excel, to allow Post Planners to check the validity of the run.

Test runs and Normal runs (N) both perform the same calculations; the major difference is that Test runs are not protected, and can be deleted from the Salary application.

## Execution

The purpose of the execution step is to perform calculations (different for different run types). The execution step will update the database with the results from these calculations. Execution in itself does not have an influence on external systems, it puts the run into an intermediary state, from where the results can be either released or removed.

### Admin Codes

The following picture shows possible transitions from one state to another.

Released

Retro

release

Cancelled

Deleted

Test

cancel

remove

remove

remove

release

[N]

retro

release

[TEST]

create

Normal

remove

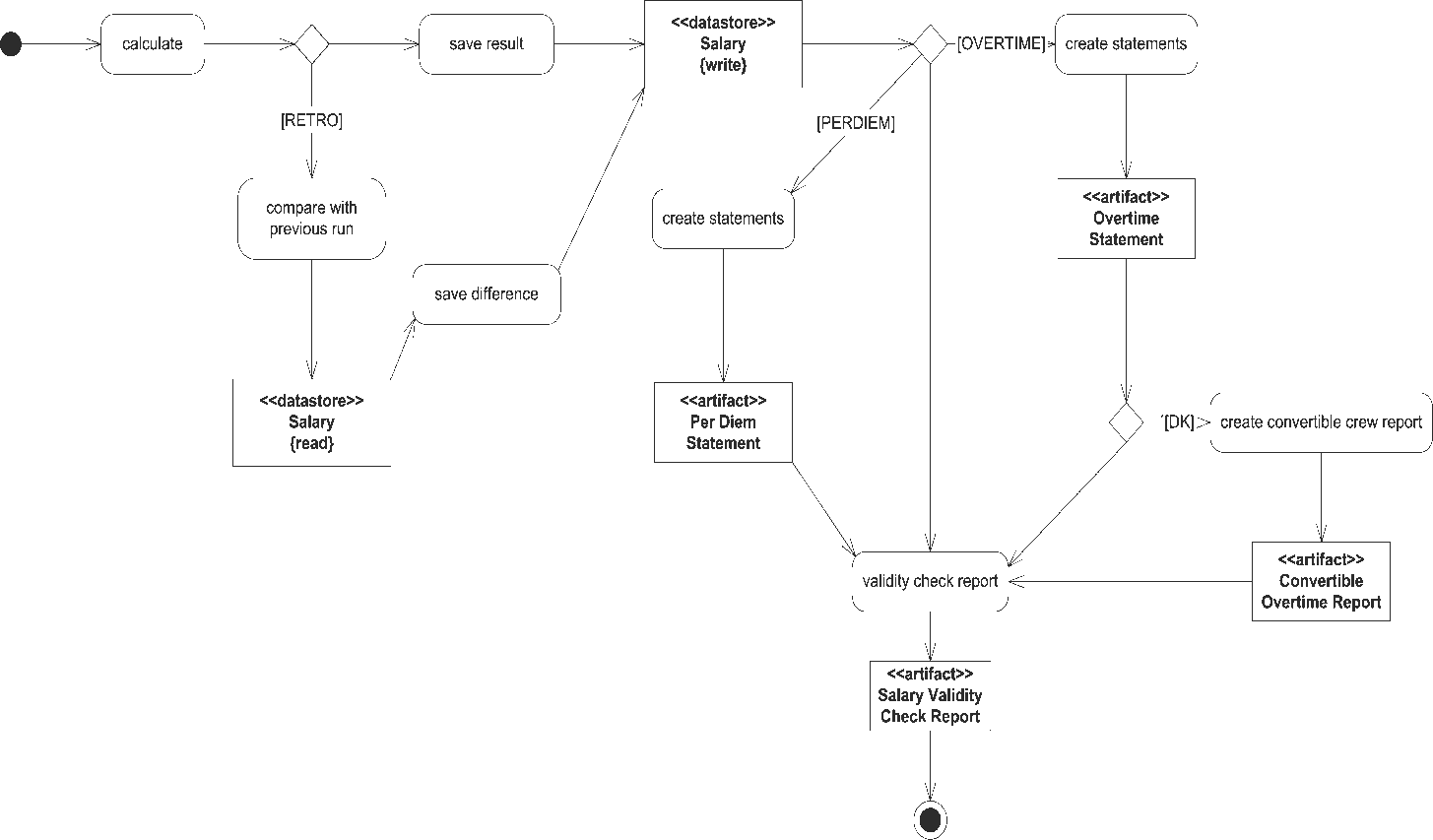
Figure

The figure shows several important characteristics:

* When a Normal job is released, then it changes state to “Released”.
* Released jobs cannot be removed.
* Test jobs cannot be released or cancelled.
* Only a job in state “Released” can be cancelled.
* It is only possible to run a Retro job based on a job that has been released.
* A Retro run implies a Normal run (N) that has been released.

### Salary Run

Salary Run will perform a number of calculations depending on the selected Run Type for each crew member and store the results in the database.



Figure

Note that a number of reports are created, depending on run type:

* All runs will produce a Salary Validity Check Report.
* Per Diem runs will produce a Per Diem statements
* Overtime runs will produce Overtime statements.
* Retro runs will produce exactly the same reports as the corresponding Normal run.

### Retro

Retro runs are used to recalculate a previous run but with current prerequisites. A retro run is created by first performing the same calculations as for a Normal run. The difference between these new values and the selected original Run are saved to the database with an entirely new Run ID. Entries where the difference is 0 (zero) will not be saved to the database to save space.

*Note: To give a crew member a full picture of the calculations requires two statements, one from the original run and one from the retro run.*

*Example:*

The Run with ID 36 had these values.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **runid** | **extperkey** | **extartid** | **crewid** | **amount** |
| 36 | 34234 | 2833 | 34234 | 2990 |
| 36 | 34235 | 2833 | 34235 | 2890 |

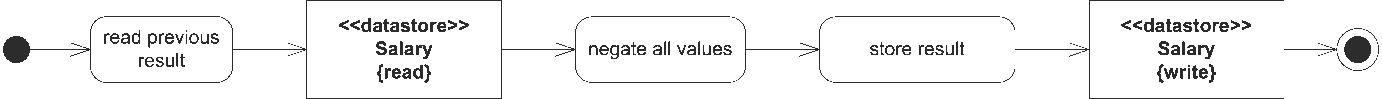
Let us say that a new calculation gives a value of 3010 for crew member 34234 and 2800 for crew member 34235.

The Retro run would add these values to the database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **runid** | **extperkey** | **extartid** | **crewid** | **amount** |
| 46 | 34234 | 2833 | 34234 | 20 |
| 46 | 34235 | 2833 | 34235 | -90 |

### Cancel

This is a special type of run that can be used to nullify (reset) a complete run. All values from the original run are written to the database using a new Run ID, but with all values negated. The old Run ID is sent as a parameter to the Cancel job.



Figure

*Note: This type of job will not create any output files.*

*Example:*

The Run with ID 36 had these values.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **runid** | **extperkey** | **extartid** | **crewid** | **amount** |
| 36 | 34234 | 2833 | 34234 | 2990 |
| 36 | 34235 | 2833 | 34235 | 2890 |

Cancel of run 36 will create these new entries in the database:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **runid** | **extperkey** | **extartid** | **crewid** | **amount** |
| 41 | 34234 | 2833 | 34234 | -2990 |
| 41 | 34235 | 2833 | 34235 | -2890 |

A new ordinary Run can be started in connection to a cancellation run; this will have the effect of nullifying the previous run and adding new (corrected) values to the system.

*Note: An effect of this is that all crew members have three values for each account on their salary specification(s), one from the first (faulty) run, a second value is a correction of the first one, and, a third value with the correct amount.*

## Validation

The purpose of the Validation step is to ensure that the calculated data is correct before making the results available for other systems or for crew.

### Balancing Report

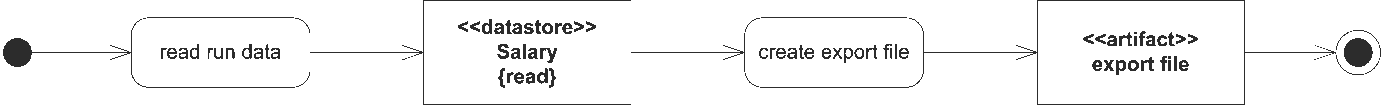
The Balancing Report (*afstemningsunderlag*) is created at release. This report can also be created and mailed to the end user by selecting Balancing Report from the Salary GUI. This report gives the big picture but does not contain any details.

### Statements and Reports

Statements and reports are checked for any incorrectness. For each account the resulting sum can be compared with values for previous periods.

### Salary Export

End users can create export files for a given run by using the Salary GUI.



Figure

*Note: No updates are made to the database.*

The export files are located in the directory: **$CARMDATA/EXPORT**.

Export files in CSV format can be created and imported into e.g. Excel where the values can be examined more thoroughly.

## Approval

The purpose of the Approval step is to make data from the Salary system available to other systems.

### Release

The Release procedure involves several steps seen in the figure below.

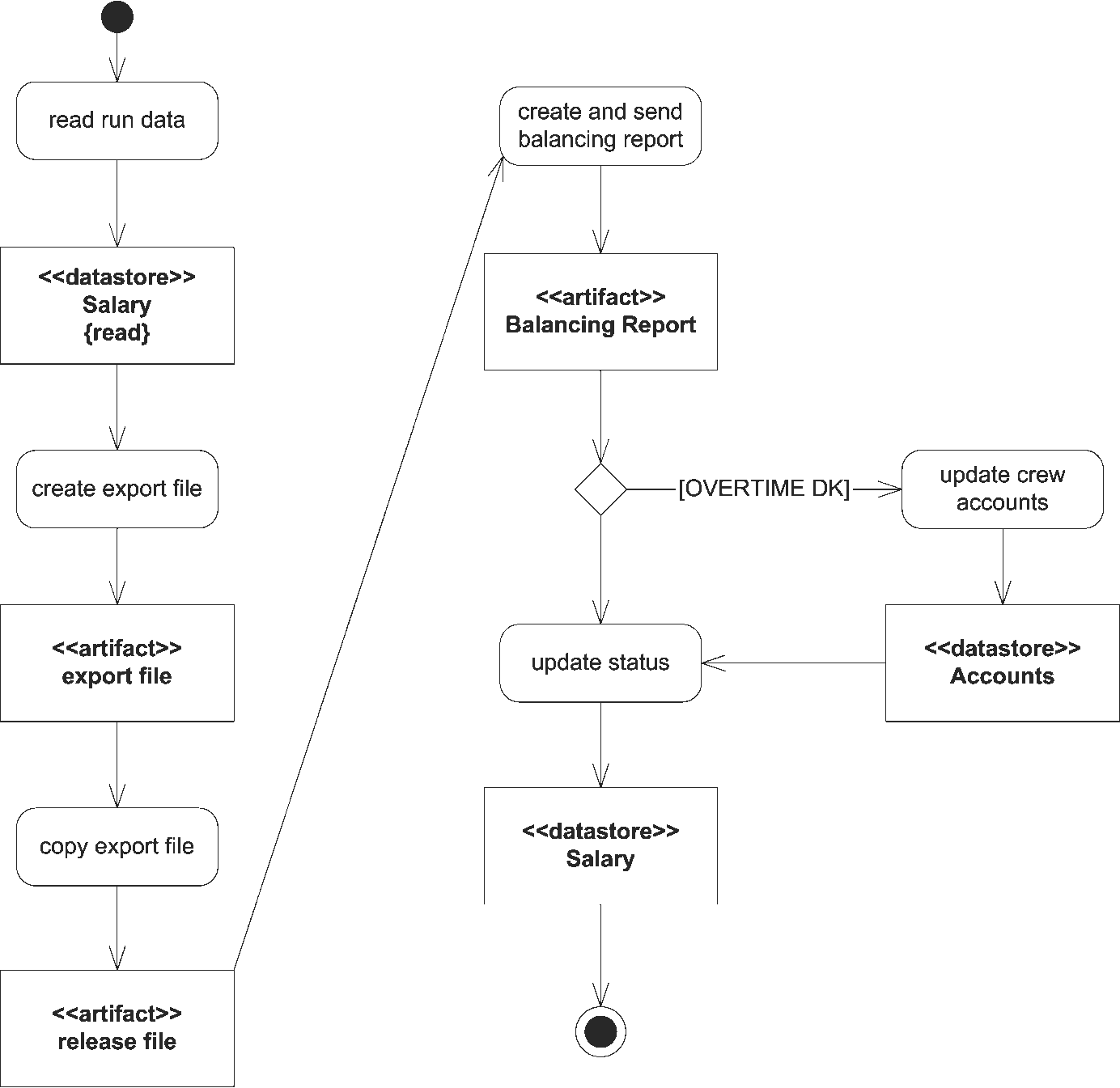


Figure 5

1. An export file (Salary file) is created and placed in a staging area, from where it can be picked up by the external salary system (PROD: /opt/Carmen/CARMTMP/ftp/out).
2. A Balancing Report (Afstemningsunderlag) is created and sent to a number of predefined email addresses found in the table: “dig\_recipients”.
3. Finally, the run is marked as completed and the release time is updated. This release flag signals that calculations are verified, approved and ready to be sent to external systems.

**Note:** In the case of Cancel (C) followed by a Normal (N) run, we need to follow this procedure:

1. Release the Cancel run.
2. Release the following Normal run.

These two steps need to be coordinated with the external system, since we can only have one released export file active at a given time! Alternatively the two files could be concatenated together to produce one single file for the external system. Either way, manual intervention is needed.

### Auto-release

The whole point of having a separate release procedure is to give salary administrators an opportunity to check the results of a run before the data is propagated to other systems and accounts are updated.

If a salary administrator is unable to perform a release, no data will be sent to external salary systems. To make sure data is released (following the principle: “wrong salary is better than no salary”), there is a separate batch job that releases the latest unreleased salary run for the given period. This job is called an *auto-release* job and is scheduled to run some time after the normal run.

These run types are currently configured to have their auto-release job scheduled on the 8th in the month, making it possible to manually release the job or re-run and then release:

* Per diem
* Instructor allowance
* Overtime
* Overtime temporary crew
* Bought days and compensation days

Some run types are considered more “reliable” than others, these run types are released immediately, and no manual intervention is needed:

* AMBI
* Vacation days performed
* Vacation days remaining
* Vacation days/year account

# Salary Application

## Prerequisites

The configuration data has to be up-to-date for the run to succeed, see 5.1 below for details.

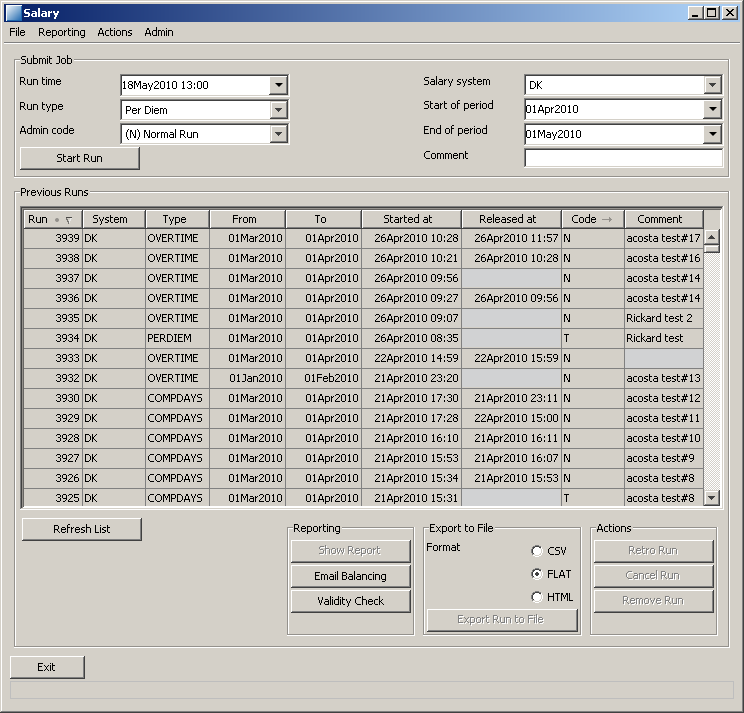
## Start Salary Application

1. Start the CMS application launcher and log on.



Figure

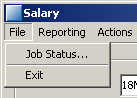
1. Click the Salary symbol to start the Salary application.



Figure

The application main window is divided into several areas. The upper part contains information that has to be submitted in order to create a new run. The middle of the window contains previous runs. The bottom part of the window contains actions that can be performed for the currently selected run (in the middle area).

### File menu



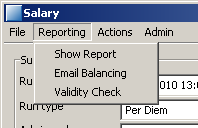
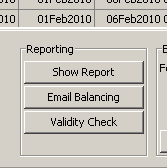
Figure

The file menu contains a couple of menu entries.

* *Job Status...* – this option will start the job viewer (see 0 below).
* *Exit* – will leave the application

### Reporting menu

The reporting menu contains actions that create different types of reports.

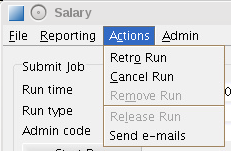
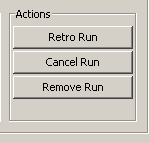
 

Figure

The buttons in the lower middle of the GUI performs the same actions as their corresponding menu entries.

* *Show Report* – pops up either a PDF report (e.g. a statement) or a HTML index file with links to the reports that were produced together with this run.
* *Email Balancing* – create a balancing report (Afstemningsunderlag) and send it by email to the current user.
* *Validity Check* – show the Salary Validity Check report.

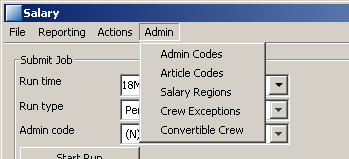
### Actions menu

Figure

This menu can be used to start actions for a selected run in the main area. Options are greyed-out if they are not applicable for the run in question. The Actions area in the lower right of the GUI performs the same actions, with one exception – the Release Run action which is only available from the menu (to avoid release by mistake).

### Admin menu



Figure

This menu contains entries for changing/showing data in configuration tables used by the Salary system. For more information see 5.1 below.

## Start Salary Run

Open the application as described in 4.2 above.

1. Set parameters, the first and last dates are the limits for the period.
2. Press *Start run* to confirm the input data and submit the job to the job queue.
3. The Job ID will be displayed on the status row.
4. When the job has finished, the run will appear in the *Previous Runs* area.

## Show Statement

Open the application as described in 4.2 above.

1. Select a run from Previous Runs.  
   *Tip: Use Ctrl+F to get a search window*
2. Press the button *Show Report*
3. A web browser window will open up with the statement*.*

*Note: This option is only available for Per Diem, Temporary Crew and Overtime.*

## Cancel Run

Open the application as described in 4.2 above.

1. Select a run from Previous Runs.  
   *Tip: Use Ctrl+F to get a search window*
2. Press the button *Cancel Run.*
3. The Job ID will be displayed on the status row.
4. When the job has finished, a new run will appear in the *Previous Runs* area.

## Retro Run

Open the application as described in 4.2 above.

1. Select a run from Previous Runs.  
   *Tip: Use Ctrl+F to get a search window*
2. Only Normal runs that have been released can be chosen.
3. Press the button *Retro Run.*
4. The Job ID will be displayed on the status row.
5. When the job has finished, a new run will appear in the *Previous Runs* area.

## Release Run

Open the application as described in 4.2 above.

1. Select a run from Previous Runs.  
   *Tip: Use Ctrl+F to get a search window*
2. Press the button *Release Run.*
3. The Job ID will be displayed on the status row.
4. The file released will be stored in the SALARY\_RELEASE folder

*Note: This option is only available for Per Diem, Overtime and Temporary Crew.*

## Send e-mails

Only available for administrators and only available on already released PERDIEM runs.  
Open the application as described in 4.2 above.

1. Select a PERDIEM run of type N (normal) or R (retro) from Previous Runs.  
   *Tip: Use Ctrl+F to get a search window*
2. Select the menu item *Send e-mails.*  
   A confirmation dialogue “Do you want to send e-mails for run XXX” is opened.
3. When Yes is selected, individual PerDiem statements are sent to crew in PDF format attached to e-mails. The e-mail addresses used are [ExtPerkey@sas.dk](mailto:ExtPerkey@sas.dk).  
   The e-mail subject will be “PerDiem Statement MMMYY - DO NOT REPLY" and the mail body will consist of a standard text "Enclosed please find you personal PerDiem Statement for MMMYY   
   Best regards SAS Crew Systems"

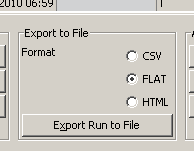
*Note: This menu item is only activated for administrators and for Per Diem jobs of type N or R and if e-mails not already sent.*

## Note about running on old periods

If you want to run a normal or a retro run for periods more than 3 months ago, DIG will automatically start up a custom report server instance. Be aware that such jobs will therefore take longer time to process than normal. After the job is finalized, the custom report server will be automatically shut down.

## Create an Export File

Open the application as described in 4.2 above.



Figure

The export file options are located in the lower right corner of the salary user interface.

1. Select a run from Previous Runs. *Tip: Use Ctrl+F to get a search window*
2. Select output format.
3. Press the button *Export Run to File.*
4. The Job ID will be displayed on the status row.

The file name of the generated export file will always be unique, to avoid that an earlier export gets overwritten by accident.

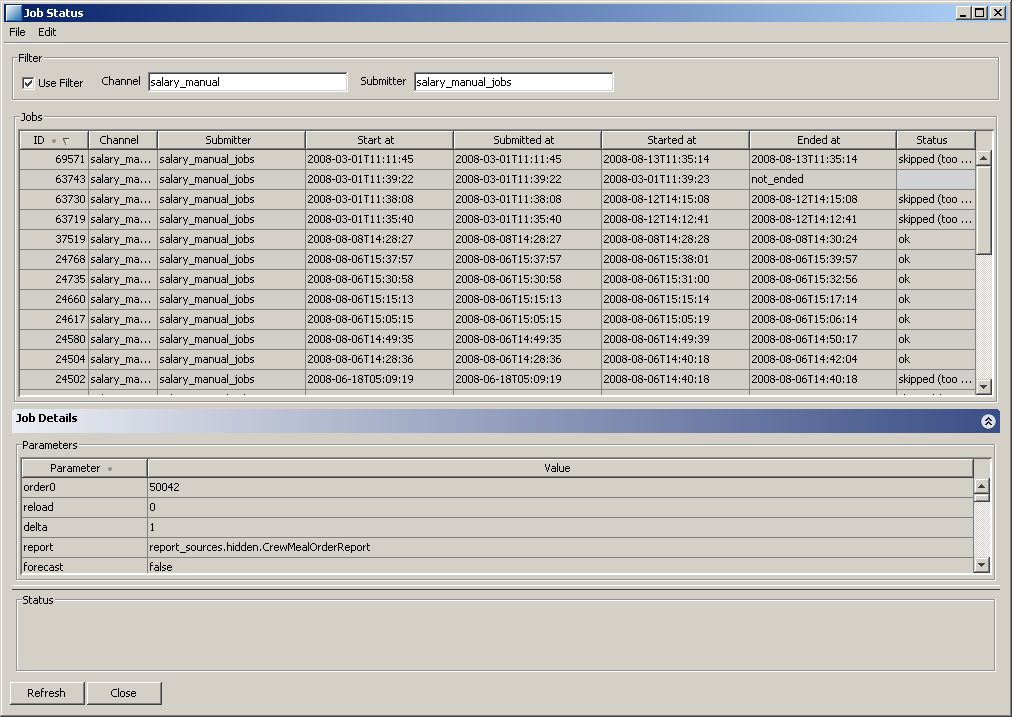
The first two positions in the file name will indicate the Salary System, then follows the Run ID and last a sequence number.

Example: **DK000008-01**, the first version of a print out for Run ID **8** and Salary System **DK**.

## Job viewer

The job viewer is used to check the status of a submitted job.

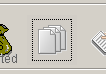
Note that there is a small delay; it can take up to five minutes (configurable in DIG) for a job to start.



Figure

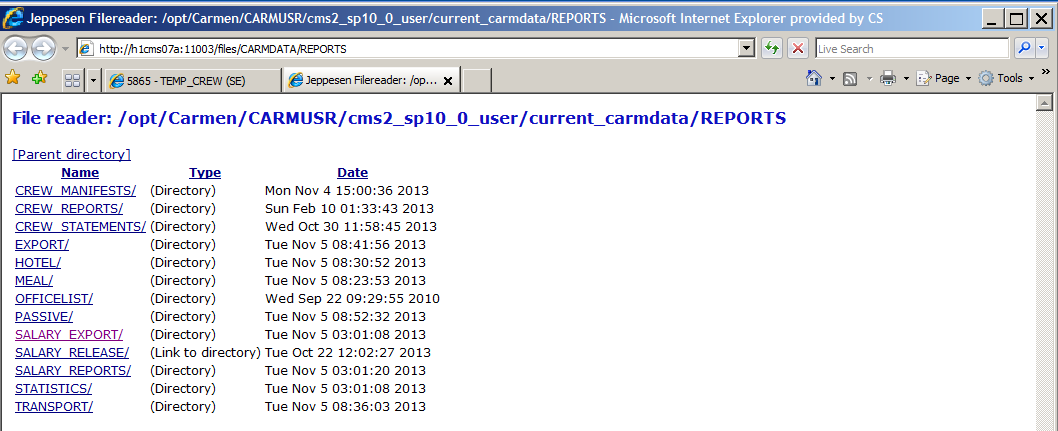
The Status field will contain a status message when the job has finished.

## Reading Export Files



Figure

Click the Filereader symbol to start Carmen Postplanner Filereader.



Select “Salary Export”.



Sort files by pressing “Date” to get latest exported files.

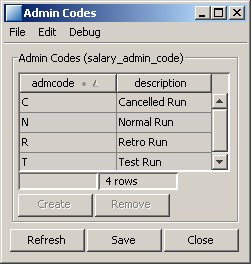
Figure

# Database layout

## Configuration tables

### salary\_admin\_code

This table contains descriptions of the different *Admin Codes*.



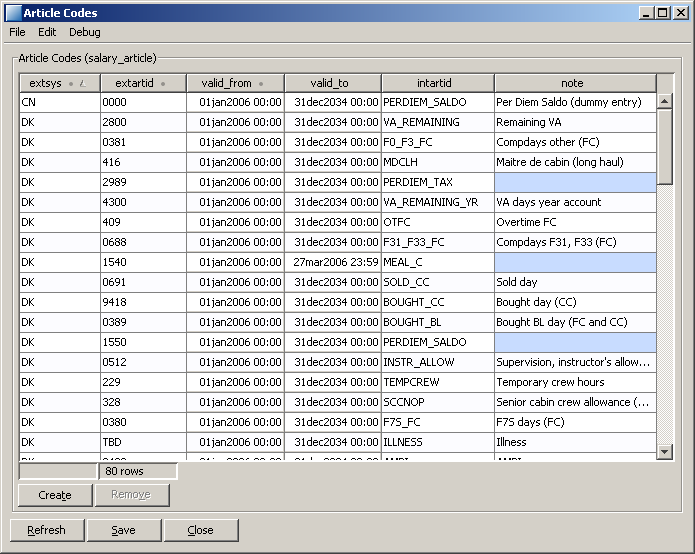
Figure

**Note:** This table should not be modified in any way.

### salary\_article

This table contains mapping between codes used internally and the different salary codes for different countries.

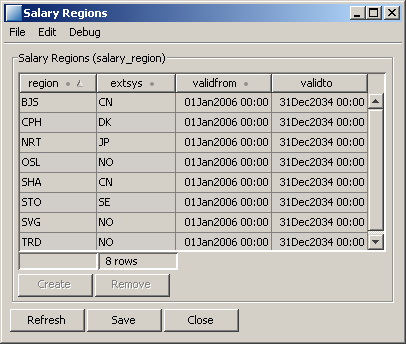
In case a code changes meaning, use the *valid\_from* and *valid\_to* fields to define the date from when the change is to take place and keep the old definition with a new expiry date. This assures that old data will be readable.



Figure

### salary\_region

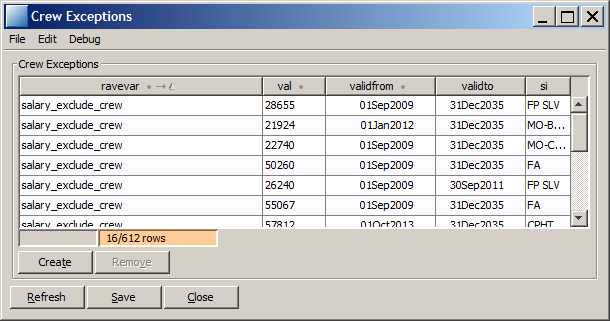
The *salary\_region* table contains mappings between home bases and salary offices. This table is not editable, since the source code has to be changed as well.



Figure

### Salary Crew Exceptions

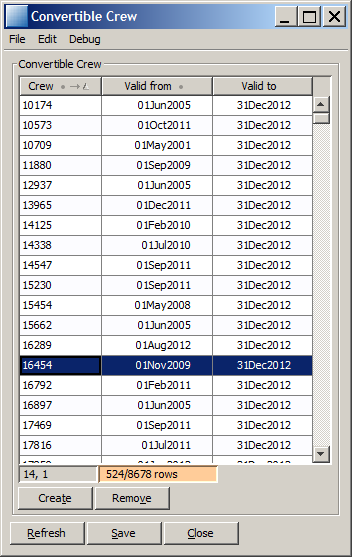
This table is used to exclude crew members from salary files and statements. The ‘val’ field should contain the internal crew ID of the crew member to exclude.



Figure

### Crew with Convertible Overtime

This table is no longer applicable.



Figure

## Data Tables

These tables are handled by the application; they should not be updated manually.

### salary\_run\_id

This table contains meta-data for each Run ID.

### salary\_basic\_data

This table contains raw data from the salary run, one row per *Run ID, Extperkey* and *Article code*.

### salary\_extra\_data

This table contains data from the salary run that are internal for CMS (intermediate results, etc.).

# Recommendations

## When to use Retro

Retro runs will contain differences between the selected *Normal* run and the current values. Only crew members with any differences between the two runs will be part of the export file. The option Retro is useful when there has been a correction of rules, e.g. new crew agreements or new PerDiem rates that affects all or group of crew.

When running Retro, please be aware that the Retro might also affect crew, that was not affected by the corrections; e.g. if flight legs or roster has been changed retroactively.

## When to use Cancel

Cancel is used to completely reset a previous run. This option should be used when the run has been considered useless and to correct the values in the salary system.

Recommendation:

In most cases you don’t have to run a Cancel, you can just run a new run, even if the previously run has been released (but not yet sent to Salary office).

You can “cancel” a normal run by changing the Code “N” (Normal run) to a “T” (Test run) in the table ‘*salary\_run\_id’* and then make a “*Remove Run*” on the “T” run. Then all records connected to that run is removed from tables, except if it is a PerDiem run, then the PerDiem Statements connected to the run has to be deleted manually from directory SALARY\_REPORTS.

If running Retro on PerDiem Norway, special considerations has to be made regarding PerDiem TAX; see Functional Reference PerDiem, § 2.4.3 Tax report Norway, last note.

End

1. Only Per Diem and Overtime, no export files are created by the batch jobs. [↑](#footnote-ref-1)