|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| PDC 1.2  User Guide | | | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  | |  | |
|  |  |  | |
|  | |  | |
|  |  |  | |

**DOCUMENT PREPARATION INFORMATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT NAME** | **PREPARED BY (PRINT)** | **SIGNATURE** | **DATE PREPARED** | **DOCUMENT ID** |
| PDC |  |  |  |  |

**CUSTOMER INFORMATION**

|  |  |  |
| --- | --- | --- |
|  | **CONTACT** | **CUSTOMER ACCOUNT** |
|  |  |  |

**GENERAL INFORMATION**

|  |  |
| --- | --- |
| **PROJECT MANAGER** | **SECURITY CLASSIFICATION** |
|  |  |

**ORIGINATOR INFORMATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **ORIGINATOR** | | **TERADATA ORGANIZATION NAME** | |
|  | |  | |
| **TERADATA ORGANIZATION ADDRESS** | | | |
| NA STRŽI 65, PRAHA 4, 140 62 | | | |
| **PHONE NUMBER** | **FAX NUMBER** | | **E-MAIL** |
|  |  | |  |

**VERSION INFORMATION**

|  |  |
| --- | --- |
| **LAST UPDATED** | **MASTER VERSION LOCATION** |
|  |  |

**CHANGE HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE** | **VERSION** | **CHANGE DESCRIPTION** | **APPROVED BY** |
| 01.02.2017 | 0.01 | Template | Teradata CZ |
| 15.02.2017 | 0.10 | Draft | Teradata CZ |
| 15.10.2017 | 2.0 | Final 1.2 | Teradata CZ |

###### Table of Contents

[1. Summary 5](#_Toc479764957)

[1.1 Abbreviations, Terms and Definitions 5](#_Toc479764958)

[1.2 Related Documents 5](#_Toc479764959)

[2. What is PDC 6](#_Toc479764960)

[2.1 Tool Component Architecture 6](#_Toc479764961)

[2.1.1 Engine 7](#_Toc479764962)

[2.1.2 Scheduler 7](#_Toc479764963)

[2.1.3 Framework checker 7](#_Toc479764964)

[3. GUI Application Layout 9](#_Toc479764965)

[3.1 GUI General Layout 9](#_Toc479764966)

[3.2 Page Layout 9](#_Toc479764967)

[3.2.1 Status Bar 10](#_Toc479764968)

[3.2.2 Main Menu 10](#_Toc479764969)

[3.2.3 Monitor Header 10](#_Toc479764970)

[3.3 Status line 11](#_Toc479764971)

[3.4 Monitor 11](#_Toc479764972)

[3.5 Control 12](#_Toc479764973)

[3.6 Logs 12](#_Toc479764974)

[3.7 SESS TABLES 12](#_Toc479764975)

[3.8 CTRL TABLES 12](#_Toc479764976)

[3.9 Admin 12](#_Toc479764977)

[3.10 Reports 12](#_Toc479764978)

[4. Initialization 13](#_Toc479764979)

[4.1 Small Initialization 14](#_Toc479764980)

[4.2 Big Initialization 14](#_Toc479764981)

[5. Monitor processing 16](#_Toc479764982)

[5.1 Monitor – Stream View 17](#_Toc479764983)

[5.2 Monitor – Job View 18](#_Toc479764984)

[5.3 Monitor – Running Jobs 18](#_Toc479764985)

[5.4 Monitor – Failed Jobs 19](#_Toc479764986)

[5.5 Monitor – Ready to Run Jobs 21](#_Toc479764987)

[5.6 Monitor – Finished Jobs 22](#_Toc479764988)

[5.7 Monitor – All Jobs 23](#_Toc479764989)

[5.8 Monitor – Odd Jobs 24](#_Toc479764990)

[6. Control processing 26](#_Toc479764991)

[6.1 Control – Scheduler 26](#_Toc479764992)

[6.2 Control – Finish Jobs 27](#_Toc479764993)

[6.3 Control – Restart Jobs 28](#_Toc479764994)

[6.4 Control – Restart Finished Jobs 29](#_Toc479764995)

[6.5 Control – Block Jobs 30](#_Toc479764996)

[6.6 Control – Unblock Jobs 31](#_Toc479764997)

[6.7 Control – Manual Batch 32](#_Toc479764998)

[7. Metadata management 35](#_Toc479764999)

[7.1 Admin – Label maintenance 35](#_Toc479765000)

[7.2 Admin – Change Management 35](#_Toc479765001)

[7.3 Admin – Access Control 36](#_Toc479765002)

# Summary

The document is a user guide for PDC product.

## Abbreviations, Terms and Definitions

| Abbreviation | Description |
| --- | --- |
| PDC | Process Dispatch Center |
| PID | Process identificator – identification of running process within operating system |
| JOB | Controled and monitored task (Teradata, SSH, ETL, CMD …) |
| STREAM | Couple of related jobs |
| STEP | Job’s part |
| ENGINE | Executive part of PDC application which lanches jobs |
| SCHEDULER | Hand-over of jobs to the Engine |
| METADATA | Storage of configuration and opareting data and procedure logic |
| GUI | Graphical User Interface for controlling and monitoring |
| CHECKER | PDC self checking application |
| SNMP | Simple Network Management Protocol |
| MANUAL BATCH | Solicit processing of some jobs |

## Related Documents

| ID | Name | Description |
| --- | --- | --- |
| 1 | PDC\_Installation\_Guide\_v1.docx | Installation step by step guide |
| 2 |  |  |

# What is PDC

Process Dispatch Centre (PDC) is controlling framework application, which is used for optimization of ETL/ELT process and its supporting tasks. The main task of PDC is launching separate jobs in sequence based on their dependencies and priorities. While doing so it utilizes optimally systems resources and uses parallelism as much as possible. The frameworks logic takes care about optimal launch of all jobs, while the internal logic of the job itself is invisible to the framework. Framework runs all jobs in the same manner – using a child process, which interprets the content of the job command line – and waits for the job exit code. In dependency on the exit code the framework considers the job finished either successfully or failed.

All necessary information needed by the framework is stored as metadata in Oracle database.

Metadata are used for:

• job configuration

• operational information about current and historical behavior of the jobs

• interface between all PDC parts

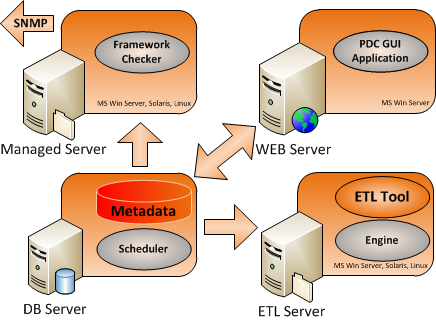
• source for representation of the current job status at GUI interface

• controlling of job behavior

There is no limitation of job type which can be executed by PDC application; the only limitation is that the interface used for job launching has to be installed on system where the framework Engine is running (e.g. for running an Informatica Job the Informatica Client need to be present on the system).

## Tool Component Architecture

The basic concept of PDC application family is shown in the picture below:



PDC application uses for its work several parts – each taking care about specific functionality:

ENGINE Executive part which launches jobs and tracks their run

SCHEDULER Hand-over of jobs to the Engine

METADATA Interdependency logic stored in Oracle procedures as well as configuration data

GUI Front end, GUI enables all necessary type of actions with PDC application.

• Monitoring - tracking progress of processing and controlling job work.

• Controlling metadata creation

• Change management support and related tasks.

CHECKER Checker detects a suspicious or erroneous situation it sends a SMTP trap to the alerting system.

### Engine

Engine represents an executive part of control framework. The main Engine responsibility is job execution and operational metadata update. The Engine launches job as independent child process which runs for whole job’s life cycle and when job has finished, update job status in metadata table. Engine is something like never ending cycle which gets launcheable jobs from Scheduler, launching them and waiting for next job for execution. Engine starts form system scheduler typically every 5 minutes, but only one Engine can run for specific engine\_id. When new instance of Engine starts, it checks the old Engine functionality. If old Engine seems to be running, the new instance stops its run, otherwise kill the process of the old Engine and takes control.

### Scheduler

Scheduler represents the brain of control framework. The procedures and tables, which are a part of Scheduler, contain almost all-processing logic. The main task of Scheduler is handing information about jobs to the Engine. Due this functionality the Scheduler is taking a care about all job related task, about parallelism control, about information delivery into all supporting application and also prepares data for Framework checker. Scheduler enables temporary stops job’s processing in case of maintenance ask. There is one instance of Scheduler for every instance of Engine.

### Framework checker

Framework checker is independent application that is responsible for checking of PDC work and alerting every irregularity. Checker is scheduled in CRON for run every five minutes. Application owns its private metadata, which contains information about typical behavior of every checked job. Application in every cycle compares current state of job with typical state and all irregularity, fail and so on reports using SMNP trap on console. Checker is looking for of these events:

• Engine stops

• Scheduler stops when some jobs is prepared for launch

• Job reaches required status

• Job doesn’t reach required status in defined time

• Job is running longer then is expected

Checker doesn’t report jobs which are running longer than expected immediately when the state appears, but allow defining some hysteresis which is protecting reporting an error on job running 6 seconds with typical run duration 5 seconds. Framework checker is not dedicated for each instance of Engine, but one Checker checks behavior of all Engines, Schedulers and jobs.

# GUI Application Layout

GUI application is WEB based application supporting standard browser functionality and using Teradata Viewpoint graphical layout. The user must have appropriate permissions for working with any part of the application. The main ask on GUI application is to enable comfortable monitoring and controlling of job processing, but GUI application is not necessary for PDC work. Jobs can be located in different instances of PDC; these instances are totally independent and are used for controlling and monitoring independent data processing. GUI shows only jobs located in selected instance of PDC but on status line are displayed statuses of all instances simultaneously. For selecting only specific part of objects a filter can be used on stream and job level. GUI consists from several parts whose meaning and functionality is described below.

## GUI General Layout

## Page Layout



Picture shows the main parties of PDC application.

The Status Bar is located on the top of page. It is used for displaying Engine’s and Scheduler’s statuses and entering to filter menu for information filtering. All displayed data is related to Engine number, which is selected in the middle of Status Bar.

Main Menu is located on the left side of the page and secondarily also under the Status Bar. Left menu can be used for direct access into menu and submenu entry. This menu can be hidden if more places for Page Related Content is necessary.

Submenu is used in case that main menu is hidden for submenu entry access.

Monitor header is the most important part of PDC application. There are shown all processing related information about jobs. More detail will be discussed later.

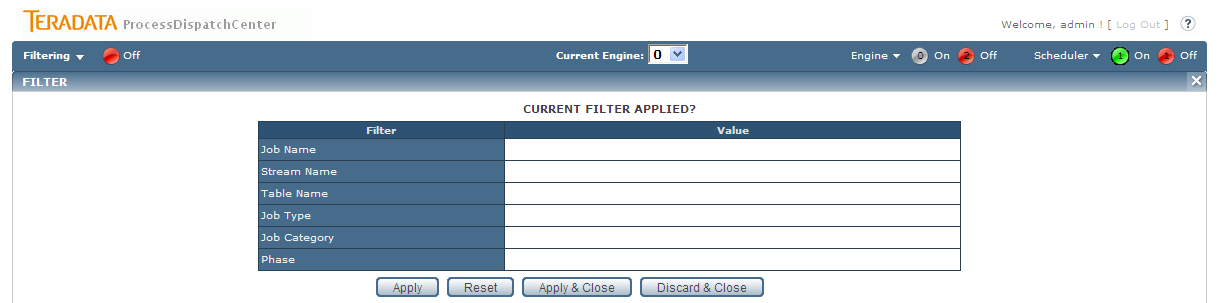
“Page Related Content” part shows chosen information that depends on menu entry, which is selected.

*Note: The default “blue” color schema can be changed for better recognition environment. Modify GUI\_COLOR parameter in CTRL\_PARAMETERS table with mean-full color name or with color hexadecimal value.*

### Status Bar

#### Filter Menu

The Filter Menu is located in left part of Status Bar. The filter works as global filter on all values related filter entry. When filter is active, only records satisfied filter condition are displayed. The search value is case sensitive and will be processed by “LIKE” %value% functionality.



#### Current Engine

There is no limitation on Engine count. Generally is supposed that Engine ID = 0 is used for regular batch processing. Therefore no Engine has any dependency on other Engine, PDC application shows only information related to Engine ID which is selected.

#### Engine

In Engine part the number of running and stopping Engine and also the Engine’s details are shown.

#### Scheduler

In Scheduler part the number of running and stopping Scheduler and also the Scheduler’s details are shown.

### Main Menu

Menu is divided into main section such as Monitoring, Control and so on. Most section contains submenu entry. Meaning of each menu entry will be discussed later. The menu content can be hidden if more places for Page Related Contend is necessary. When main menu is hidden you can still touch entry in Menu and Submenu part on top part of the page.

### Monitor Header

Monitor header shows the most important information about batch processing. This information is displayed:

* Environment – the environment kind which is monitored such as PROD, TEST or DEVL. There is no limitation on number or kind of environments but please note that every environment needs separate instance of metadata database and GUI application. Shown value is taken from CTRL\_PARAMETERS table.
* Load Date – load date or load timestamp for intermediate load is displayed
* Task Type – AUTOMATIC for regular batch processing is shown. MANUAL is displayed when Manual batch is in progress.
* Provided by – in case of Manual batch processing the user name that initialized Manual batch is shown, otherwise SCHEDULER is displayed.
* Task(s) – maximal number of running jobs concurrently is shown.
* Refresh Rate – refresh rate in second of information placed on Monitor Header is displayed
* Running – number of currently running jobs is displayed. Number works as hyperlink to corresponding page which shows job’s details of running jobs.
* Failed Jobs – number of failed jobs is displayed. Please note that it’s big difference between Failed jobs and jobs in failed status. There are only placed jobs which already reach maximal number of automatic restart. These jobs have to be restarted by supervisor. Number is hyperlink to corresponding page which shows job’s details.
* Ready to Run Jobs – number of jobs not launched yet. Number is hyperlink to corresponding page which shows job’s details.
* Finished Jobs – number already finished jobs. Number is hyperlink to corresponding page which shows job’s details.
* Odd Jobs – number of jobs running oddly is displayed. The main reason for job become oddly is that job is running out of statistics. Number is hyperlink to corresponding page which shows job’s details.

Tip: The best guidance for supervising is holding values in:

* Running – the value grater then zero means that some jobs are currently running, zero value, when job’s processing is not finished yet, is unwanted.
* Failed Jobs – when not zero value is displayed, the job’s problem has to be solved
* Odd Jobs – not zero value says that some jobs are running out of statistics, the potential job’s freezing has to be revised.

## Status line

Status line is located in the top level of GUI application and shows the number of the engine which is selected for monitoring and controlling. Statuses of other Engines and Schedulers are also displayed. For faster touching of information which is looked for a filter can be used.

## Monitor

Monitor is used for displaying of current status of processing. The information can be presented from stream or job point of view. Drill down functionality enables drill for further details, so from stream view the user can drill down the information of how many jobs are located in the stream, what they are and what their status is. User can also drill down to parameters of selected job. On job level the user operates the job; it means he can abort running job, restart or finishing failed job and so on. Jobs and streams are divided into processing classes which represents objects state such as prepare for run, running, finished, failed and so on. The environment, load date and task type is shown as well as display refresh rate and maximal number of concurrently running jobs. The status overview part contains number of currently running jobs, number of failed jobs, number of jobs prepared for run and number of already finished jobs shows. All these numbers support drill down functionality, it means you can directly get a list of jobs in a category by clicking on the appropriate number.

## Control

Control enables access to all necessary settings used for controlling of PDC application. Same basic functionality for job processing is also located on the monitor part, but Control part enables driving parallelisms, temporary stop the Scheduler or doing control task on job level simultaneously for group of jobs. User can also simply stop executing jobs on selected dependency branch by blocking its parent job. Manual Batch functionality enables recalculation of selected jobs for chosen load date. It’s often used for datamarts recalculation for the day.

## Logs

Logs display controlling steps done on PDC application – it enables potential supervisor cooperation on problem solving, and also all problems captured by Framework checker application.

## SESS TABLES

SESS (session) tables contain operational metadata and store actual status of objects. SESS TABLES part enables changes in the table content. Such change can represent increasing actual job priority, command line changes and so on.

## CTRL TABLES

CTRL (control) tables store configuration metadata of jobs, streams, their dependences, locks and calendars. This part of GUI application is used for creating or changing this metadata. Application automatically checks correctness of typed values. Changes are enabled only when a label is selected which is necessary for change management process package creation.

## Admin

Admin section contains necessary administrative tasks such as label maintenance, change management process, access control and so on.

## Reports

Reports page is a signpost for customer reports related for PDC processing. Customers can simply use their reporting tools for creating reports based on PDC metadata and place them on this page.

# Initialization

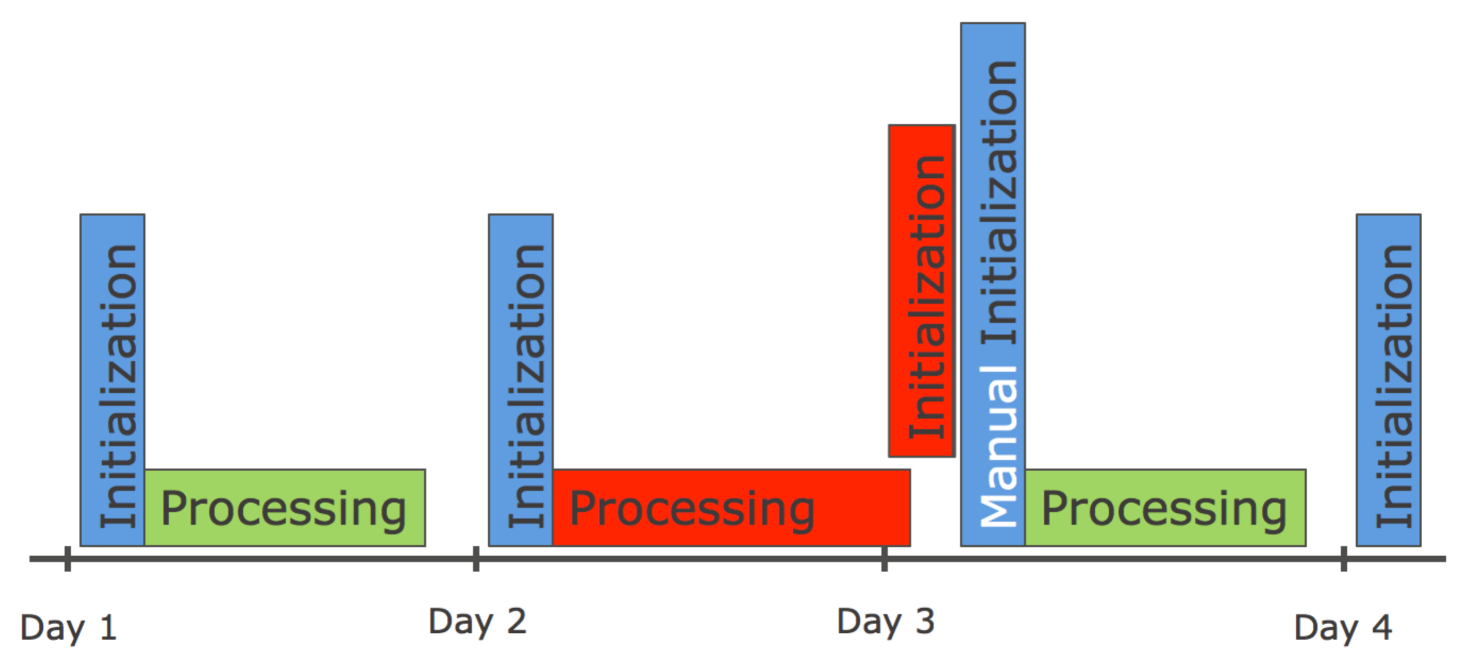
Initialization is used for setting jobs into initial state. Control metadata are taken from tables:

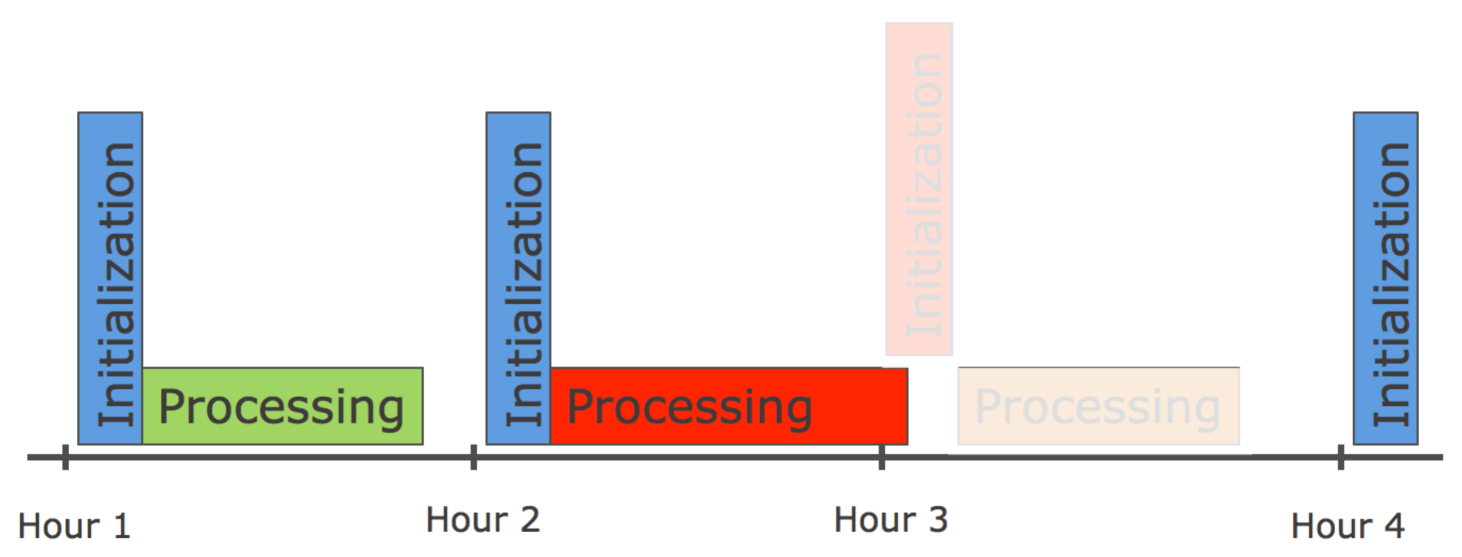
* CTRL\_STREAM – stream definition – stream\_begin and stream\_end jobs definition
* CTRL\_JOB – job parameters
* CTRL\_STREAM\_DEPENDENCY – stream to stream dependency and dependency type
* CTRL\_JOB\_DEPENDENCY – job to job dependency and dependency type
* CTRL\_STREAM\_PLAN\_REF – calendar for job’s run definition on stream level
* CTRL\_JOB\_TABLE\_REF – table lock definition on job level

Data are stored into tables:

* SESS\_JOB – current state of jobs
* SESS\_JOB\_DEPENDENCY – job to job dependency (parent – child)

There are two steps of initialization. In the first step (small initialization) only that jobs which provided the initialization work are initialized. These jobs can be simply recognized therefore they have phase = INITIALIZATION. In the second step (big initialization) Engine runs the small initialization jobs that are doing all necessary steps for initialization.





Two pictures above discuss daily and inter-day initialization difference. Every initialization is process, which is scheduled in OS scheduler. Initialization process itself has integrated check for testing if all jobs from the last processing are already finished. If not, initialization cannot continue. On the first picture the situation in daily load is drawn. Initialization of Day-3 cannot be done due running jobs from Day-2. It is necessary to correct the situation by manual launching of initialization when old day processing is finished. On the second picture the inter-day processing is shown. There is no chance for manual launch of Initialization when the last period jobs are running in time of initialization. In this case Hour-3 jobs are not initialized, but initialization time is updated. Factually it means that there is no Hour-3 processing, Hour-4 processing will have to process the source data together with its data. The initialization behavior is defined in CTRL\_PARAMETERS table by “INITIOLIZATION\_MUST\_RUN” value. For daily processing value 1 guarantee that initialization process will not be skipped, otherwise value 0 for inter-day processing allows unbreakable run in case when jobs are still running in initialization time.

## Small Initialization

All functionality of small initialization is in procedure SP\_INIT\_PREPARE which is called form PERL script periodically from CRON on weekly, daily or hourly bases. Procedure provides these steps:

1. Check if all jobs for engine\_id are already finished. If not, initialization is skipped therefore operational metadata should be damaged in case when initialization will continue. If load is marked as interday load some initialization steps are provided, such as LOAD\_DATE update. For not interday load no steps are provided, initialization has to be run manually once more when all jobs are finished.
2. CURRENT\_TIMESTAMP of initialization begin is stored into INITIALIZATION\_BEGIN parameter in CTRL\_PARAMETER table.
3. All jobs for engine\_id are deleted from SESS\_JOB table
4. All records for not existing child or parent jobs in SESS\_JOB table are deleted from SESS\_JOB\_DEPENDENCY table
5. If value of INITIALIZATION\_IS\_RUNNING parameter (in CTRL\_PARAMETER table) equal zero, value one is set and next steps are executed, otherwise value of parameter is left and next steps are skipped (value of parameter = 1 means that the last initialization wasn’t finished successfully)
   1. Value of LOAD\_DATE parameter is stored in PREV\_LOAD\_DATE
   2. Value of INITIALIZATION\_RETENTION\_PERIOD is added to LOAD\_DATE parameter value (next load date is set)
6. Value of APPLICATION\_ID parameter for engine\_id changed to be one (initialization)
7. Scheduler is switched off by setting value of MAX\_CONCURRENT\_JOBS to zero for engine\_id.
8. SESS\_JOB\_DEPENDENCY table is populated by dependency records of small initialization jobs
9. Jobs INITIALIZATION\_STREAM\_BEGIN and INITIALIZATION\_STREAM\_END are created in SESS\_JOB table.
10. All jobs from CTRL\_JOB table having phase = ‘INITIALIZATION’ are transferred into SESS\_JOB table
11. Scheduler is switched on by changing value of parameter MAX\_CONCURRENT\_JOBS. The minimum value from MAX\_CONCURRENT\_JOB\_SET and MAX\_CONCURRENT\_JOBS\_DFLT parameter is taken.

Next Engine starts small initialization jobs which causes that big initialization is provided.

## Big Initialization

Big initialization is constitutes by small initialization jobs. The small initialization jobs can have different meaning, but typically they have to do:

* Initialization of all processing jobs
* Job’s statistics update
* Other special work execution like logs archiving or tables from UTILITY database archiving
* Initialization end confirmation

#### Job’s initialization

Initialization has to initialize all jobs for engine\_id from CTRL\_JOB table which are not small initialization (phase <> ‘INITIALIZATION’). These jobs can’t be placed into SESS\_JOB table directly therefore Engine is running so all jobs has to be placed into SESS\_JOB\_BCKP table first. All functionality is written in SP\_INIT\_INITIALIZE procedure which is called from PERL script. Procedure provides these steps:

1. Distinct list of runplan values is taken from CTRL\_STREAM\_PLAN\_REF table and list is placed into TEMP\_PLAN table.
2. Value of initial job’s status is calculated for every record in TEMP\_PLAN\_TABLE.
3. Jobs STREAM\_BEGIN and STREAM\_END are created in SESS\_JOB\_BCKP table for all stream from CTRL\_STREAM table which is initialized
4. All regular jobs which is initialized from CTRL\_JOB table are placed into SESS\_JOB\_BCKP table
5. Initial value of job’s status is taken from TEPM\_PLAN table
6. All dependencies are built in SESS\_JOB\_DEPENDENCY\_BCKP table
7. Also dependency of all jobs on INITIALIZATION\_STREAM\_END job which blocked launching jobs before all small initialization jobs are finished is placed
8. Job’s initial status is modified by status\_begin value for all jobs with not null value of status\_begin
9. All records from SESS\_JOB\_DEPENDENCY\_BCKP table is placed into SESS\_JOB\_DEPENDENCY table
10. Finally all records from SESS\_JOB\_BCKP table is placed into SESS\_JOB table

#### Job’s statistics update

Job’s statistics has to be updated for possibility to compare current job’s behavior with standard one. Functionality is written in SP\_INIT\_RECALC\_STATISTICS procedure that has these steps:

1. Records from SESS\_STATUS table are used for update of SESS\_JOB\_STATISTICS table:
   1. Avg\_duration value is calculated as difference in seconds between last\_start\_ts and end\_ts for successfully finished jobs
   2. Avg\_end\_tm value is calculated as difference in seconds between end\_ts and timestamp when initialization has finished (INITIALIZATION\_END in CTRL\_PARAMETERS table)
2. Records from SESS\_JOB\_STATISTICS are inserted into STAT\_JOB\_STATISTICS, already retired records are deleted
3. Content of SESS\_STATUS table is backuped into STAT\_STATUS table and deleted
4. Estimates are placed into SESS\_JOB\_STATISTICS table:
   1. Value of day\_in\_week is and day\_in\_month is taken from PREV\_LOAD\_DATE parameter, Monday = 1, Sunday = 7, Ultimo of the month = 999
   2. Rules used for avg\_duration and avg\_end\_tm calculation:
      1. Values for day\_in\_week and day\_in\_month are calculated separately
      2. Only last 15 values are taken
      3. Average is calculated from taken values
      4. The higher value of day\_in\_week and day\_in\_month is taken

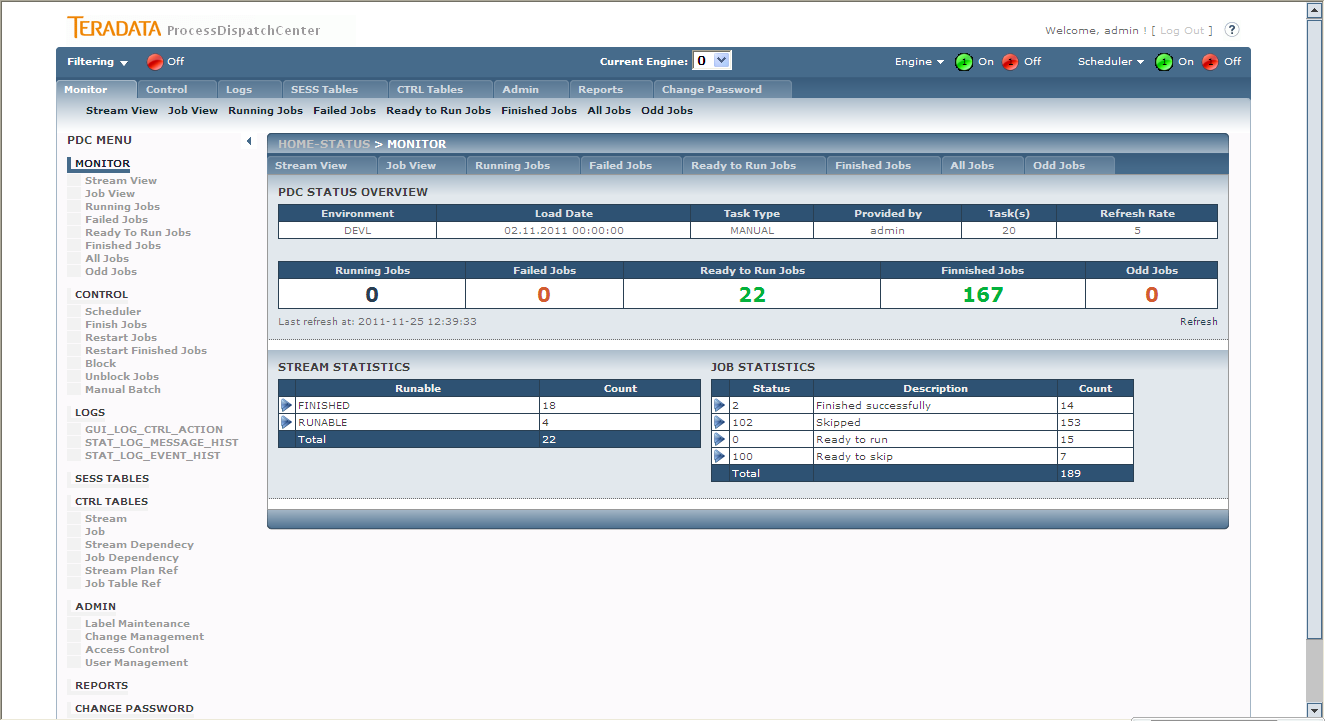
#### Initialization end confirmation

Job initialization end confirmation has to be the last initialization job. Values of some parameters in CTRL\_PARAMETERS table are changed. Oracle procedure SP\_INIT\_INITIALIZATION\_END is called from PERL script Init\_Initialization\_end.pl. These parameter’s values are changed by procedure:

1. APPLICATION\_ID is changed to 0 (regular processing).
2. INITIALIZATION\_END is populated by current timestamp
3. INITIALIZATION\_IS\_RUNNING is set to 0

# Monitor processing

Monitor is used for displaying of current status of processing. Information can be presented from stream or job point of view. Drill down functionality enables drill for further details, so from stream view the user can drill down the information of how many jobs are located in the stream, which jobs they are and what their status is. User can also drill down to parameters of selected job. On job level, the user can operate the job; it means he can abort running job, he can restart or finish failed job and so on. Jobs and streams are divided into processing classes depends on its status. The class contents objects state such as prepare for run, running, finished, failed and so on. The environment, load date and task type is shown as well as display refresh rate and maximal number of concurrently running jobs. The status overview part shows number of currently running jobs, number of failed jobs, number of jobs prepared for run and number of already finished jobs. All these numbers support drill down functionality, it means you can directly get a list of jobs in a category by clicking on the appropriate number.

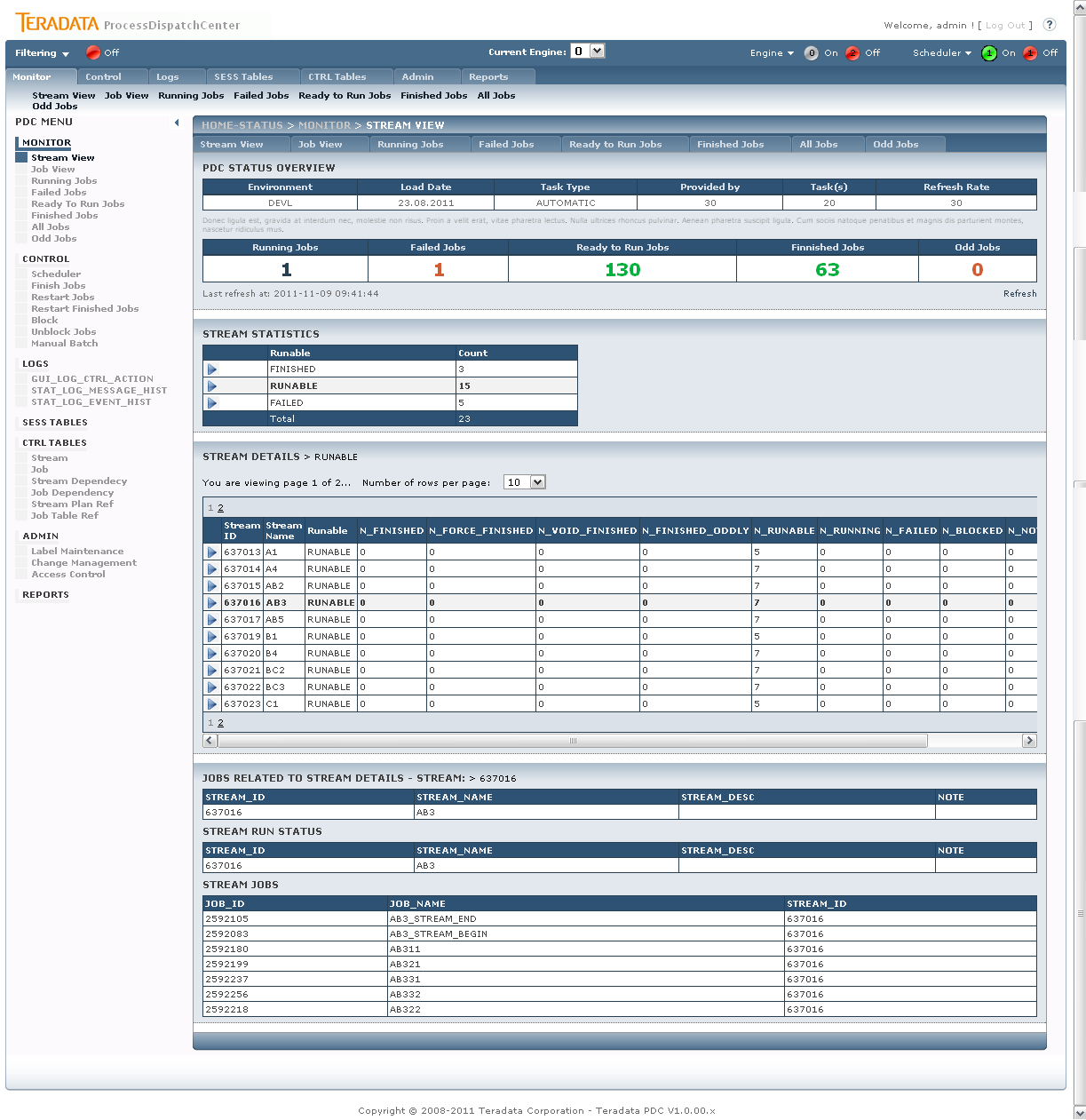


Picture shows the basic Monitor page. Only Page Related Content will be discussed, all other parts were already discussed above. There are two different lookup possible on processing – the stream related view and job related view. In both parties the number of jobs or streams in defined status is displayed. From stream perspective there are these status categories available:

* FINISHED – all jobs in stream are already finished
* FORCE\_FINISHED – all jobs in stream are already finished, some jobs were force finished (job’s task wasn’t realized)
* VOID\_FINISHED – all jobs in stream are already finished, some jobs failed but successfully finishing of these jobs weren’t required
* FINISHED\_ODDLY - all jobs in stream are already finished, some jobs finished oddly
* RUNABLE – some jobs of stream weren’t launch yet
* RUNNING – some jobs of stream are currently running
* FAILED – some jobs of stream are in failed status
* BLOCKED – some jobs of stream are manually protected to launch

From job view there are many statuses which saying in which state job currently appears. The meaning of statuses is similar like statuses of the stream. Several statuses have same meaning even if number differs therefore status “remember” history of job’s processing.

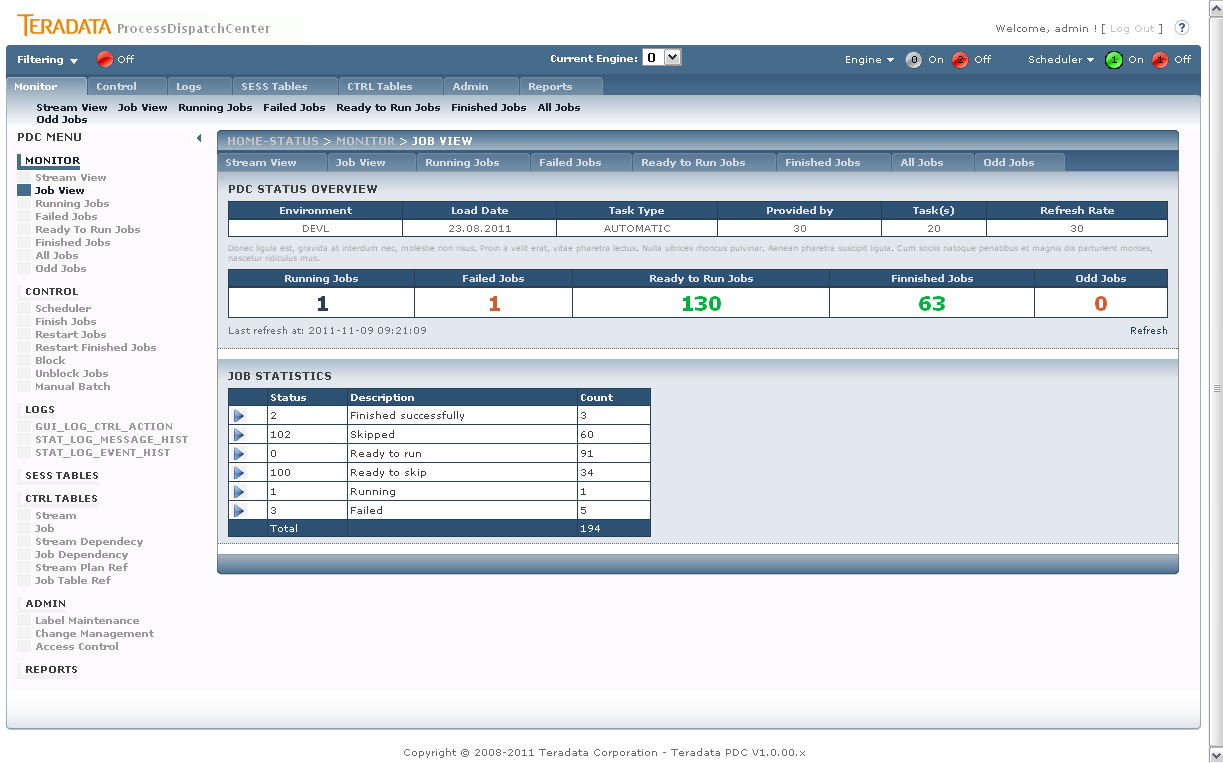
## Monitor – Stream View



Picture shows Monitor – Stream view with details

All status categories in stream view are clickable. Clicking on it, the stream list assigned into this category appears. Detail table shows number of jobs in each status category for every stream. Clicking on stream the list of jobs assigned to this stream appears.

## Monitor – Job View

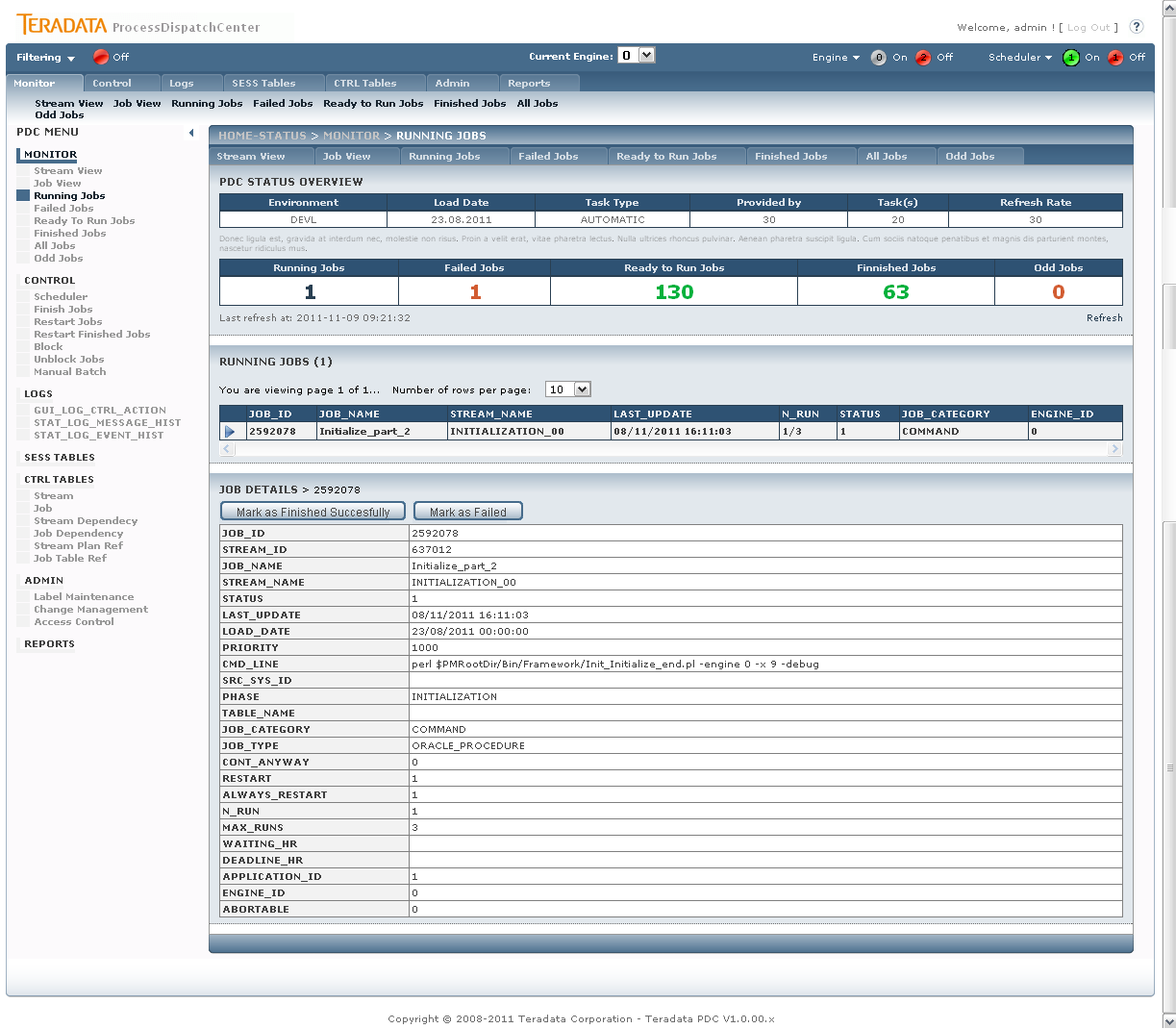


Picture shows the Monitoring – Job View page.

Monitoring – Job View page shows separate statuses and number of jobs currently having this status. All statuses are clickable. When clicking on it, the list of jobs in status appears. Clicking on job displays job’s details. This section is identical with detail’s section of the next monitoring pages and will be discussed in this section.

## Monitor – Running Jobs

There are two possibilities how to reach this page. The first is using Running Jobs entry in menu, the second one is clicking on Number of Running Jobs in Monitoring Header section.



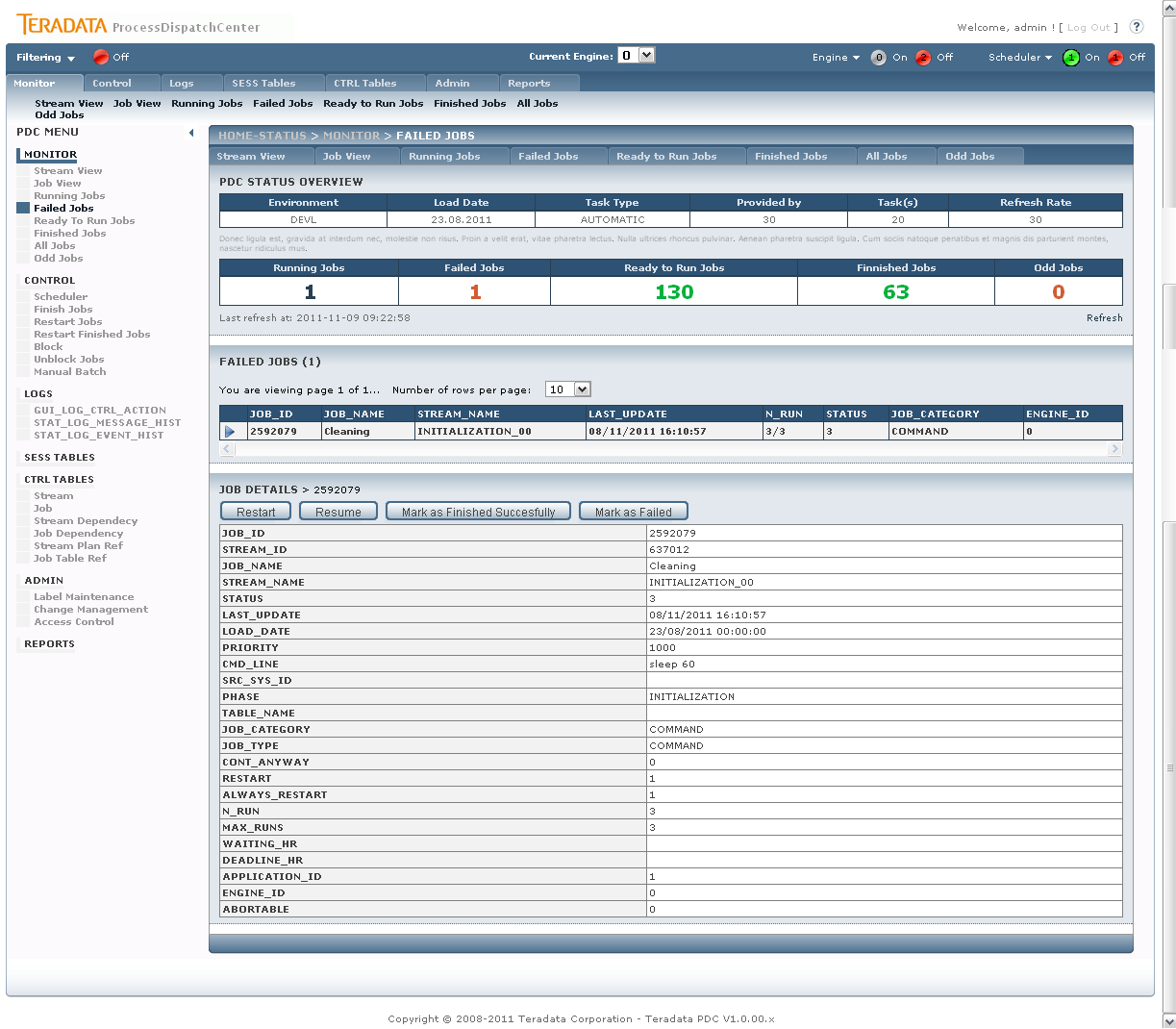
Picture shows Monitoring – Running Jobs with details

Monitoring – Running Jobs displays the job list which is currently running. Please note that there is no automatic refresh of job’s details due ensure comfortable work with them, so the list couldn’t be current after some piece of time. Clicking on job the job’s details from SESS\_JOB table appears. On the top part of job’s detail there are buttons for job operation. Buttons have this functionality:

* Abort Job – only when job is abortable is displayed. Clicking on this button, the special job for aborting of running job is started.
* Mark as Finished Successfully – marked jobs as successfully finished. This functionality is used only in case when job finished successfully but job’s metadata from oddly reason aren’t updated.
* Mark as Failed – marked jobs as failed. This functionality is used only in case when job failed but job’s metadata from oddly reason aren’t updated.

## Monitor – Failed Jobs

There are two possibilities how to reach this page. The first is using Failed Jobs entry in menu, the second one is clicking on Number of Failed Jobs in Monitoring Header section.



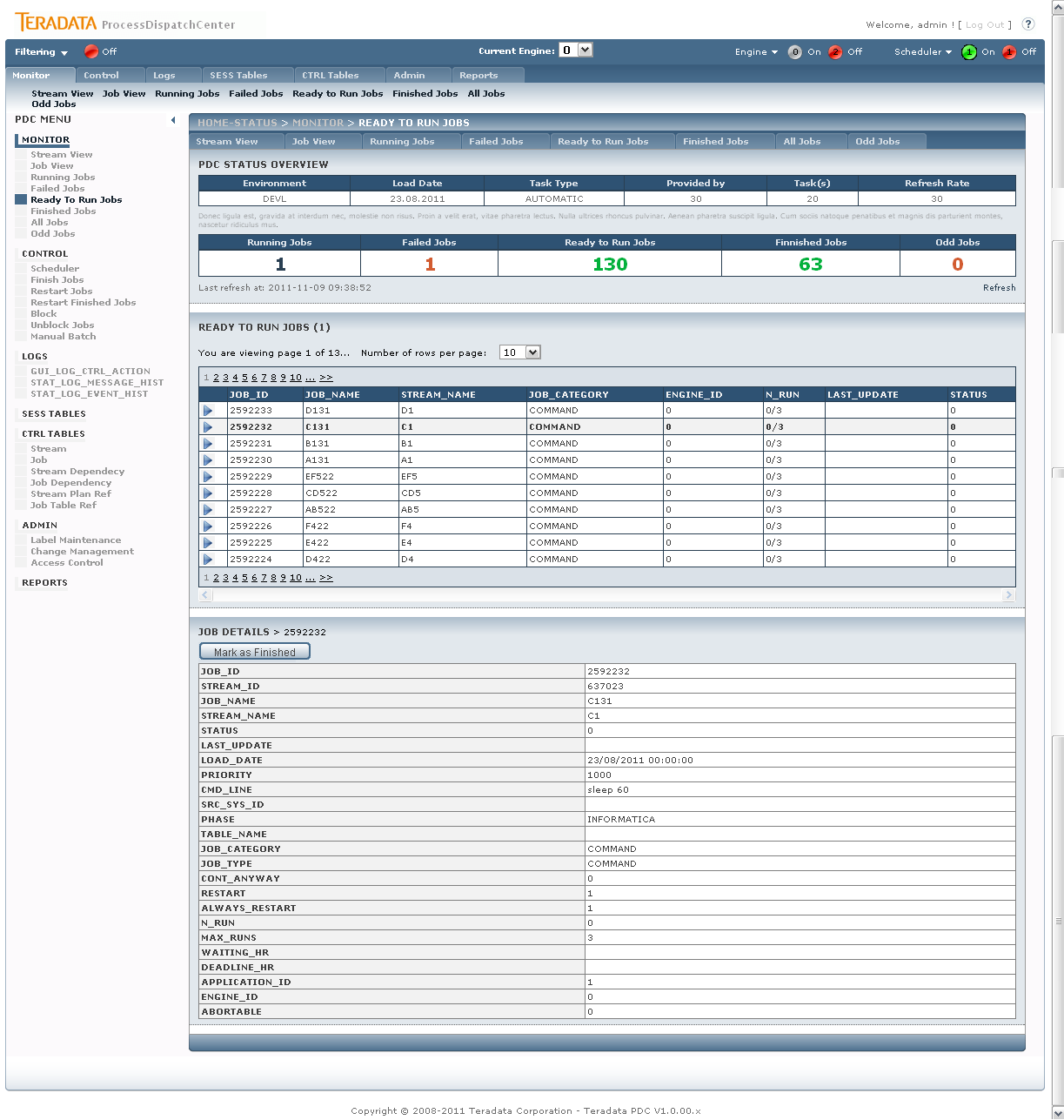
Picture shows Monitoring – Failed Jobs page with details

On this page there are only jobs which are in status fail and already reach the number of automatic restart (n\_run = max\_runs) displayed. These jobs have to be operated by supervisor. On the top part of job’s detail page there are buttons for job operation. Buttons have this functionality:

* Restart – job is returned to Scheduler by increasing value of max\_runs parameter by the value of max\_runs parameter from CTRL\_JOB table. The value of restart parameter is set to true even if resume of job is permited.
* Resume – job is returned to Scheduler by increasing value of max\_runs parameter by the value of max\_runs parameter from CTRL\_JOB table. The value of restart parameter is set to false only when value of always\_restart is set to false also. It means that resume has the same functionality as restart for jobs with value of always\_restart parameter set to true.
* Mark as Finished – sending the signal for force finishing job. The job is removed from processing without its successfully ending, child jobs can be launched.

## Monitor – Ready to Run Jobs

There are two possibilities how to reach this page. The first is using Ready to Run Jobs entry in menu, the second one is clicking on Number of Ready to Run Jobs in Monitoring Header section.



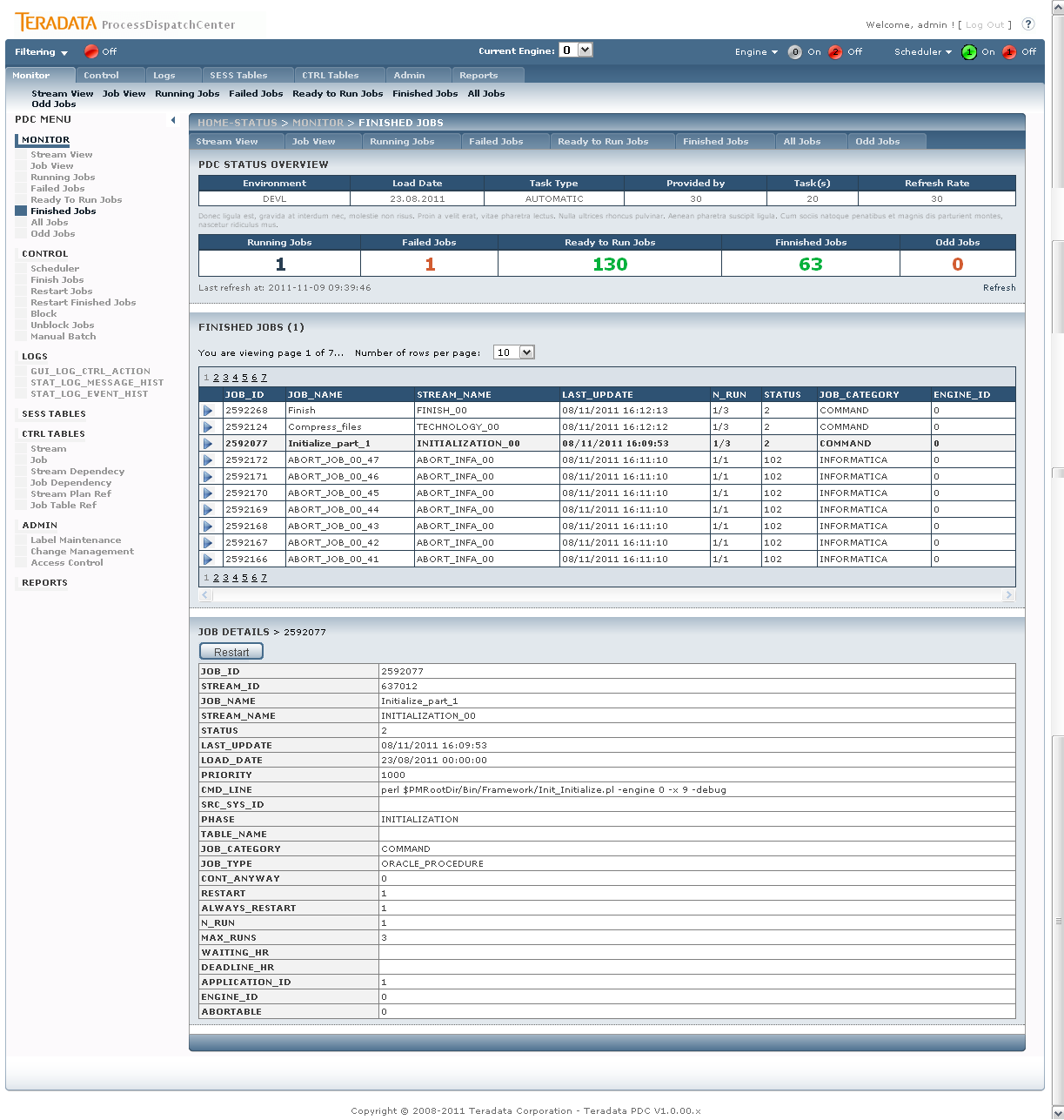
Picture shows Monitoring – Ready to Run Jobs with details

There are shown jobs prepared for launch on this page. It doesn’t matter if jobs are really prepared, it means having no dependency, no conflict with currently running or failed jobs and so on, or not. On the top part of job’s detail there is button for job operation. Button has this functionality:

* Mark as Finished – this button releases the job from processing. Factually the job can’t be removed therefore ending of the job causes removing its dependency. From this reason only status of the jobs is changed and job is finished by Scheduler when it comes into processing.

## Monitor – Finished Jobs

There are two possibilities how to reach this page. The first is using Finished Jobs entry in menu, the second one is clicking on Number of Finished Jobs in Monitoring Header section.



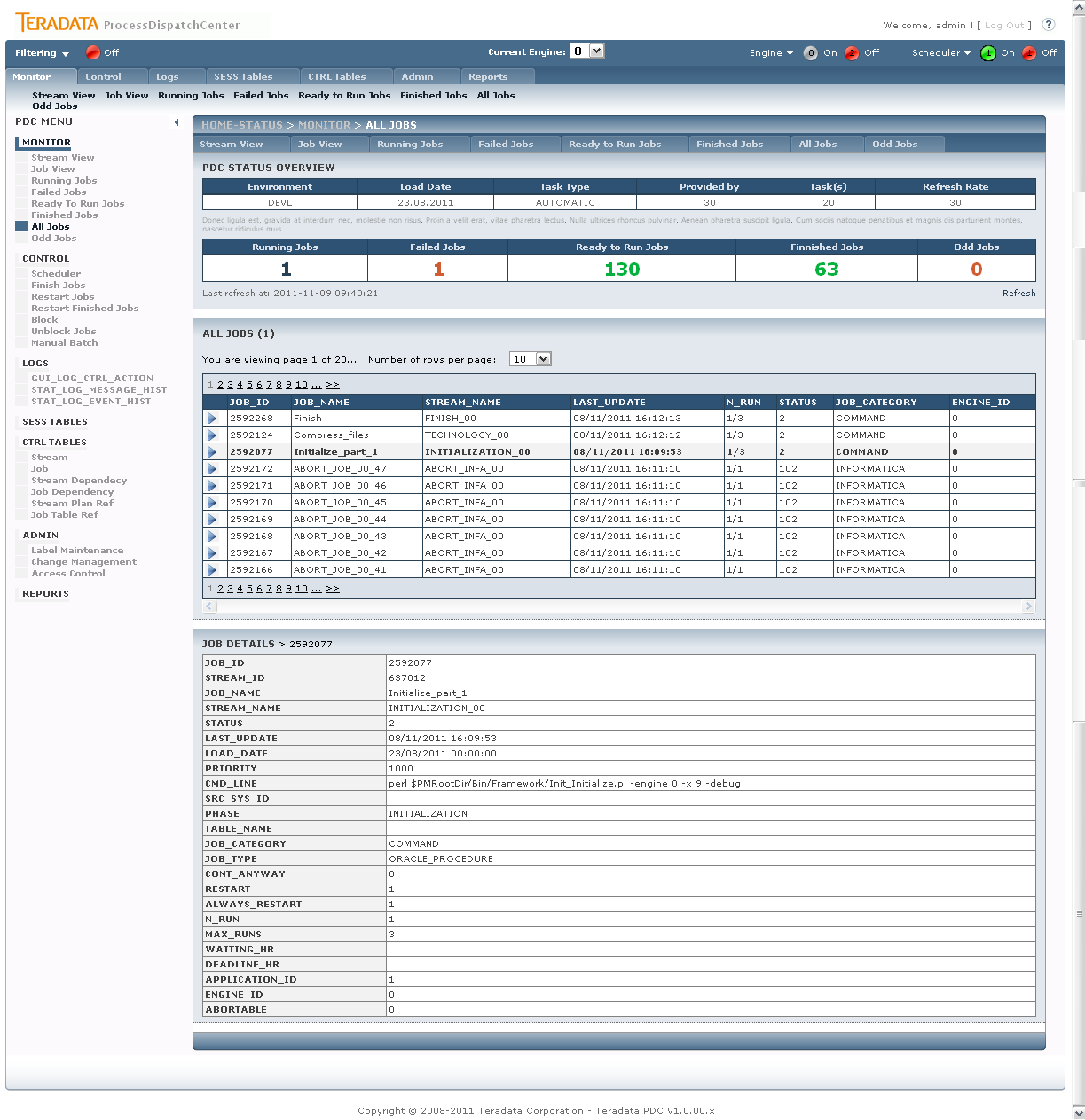
Picture shows Monitoring – Finished Jobs with details

There are shown jobs already finished on this page. On the top part of job’s detail there is button for job operation. Button has this functionality:

* Restart – job is returned to Scheduler to be launched again. It is difference between Restart here and Restart in Failed Jobs section. Here already finished job is asked to be restarted, so it is not enough only increasing value of n\_run parameter, but also status of the job has to be changed from finished to ready to run. Please remember that there are no job dependency in time of job restart, so it is not possible simply restart parent and child job together therefore Scheduler doesn’t know anything about execution order of these jobs now. If this task is required, you have to restart parent job first and wait until it finished and then restart child job. For comfortable processing of group of jobs, only Manual Batch is preferred to be used.

## Monitor – All Jobs

This page can be reached only from menu entry therefore there is no special meaning of the jobs listed on this page. This page is used in advance for searching within jobs when you are looking for job’s information and you have no knowledge about current job status.

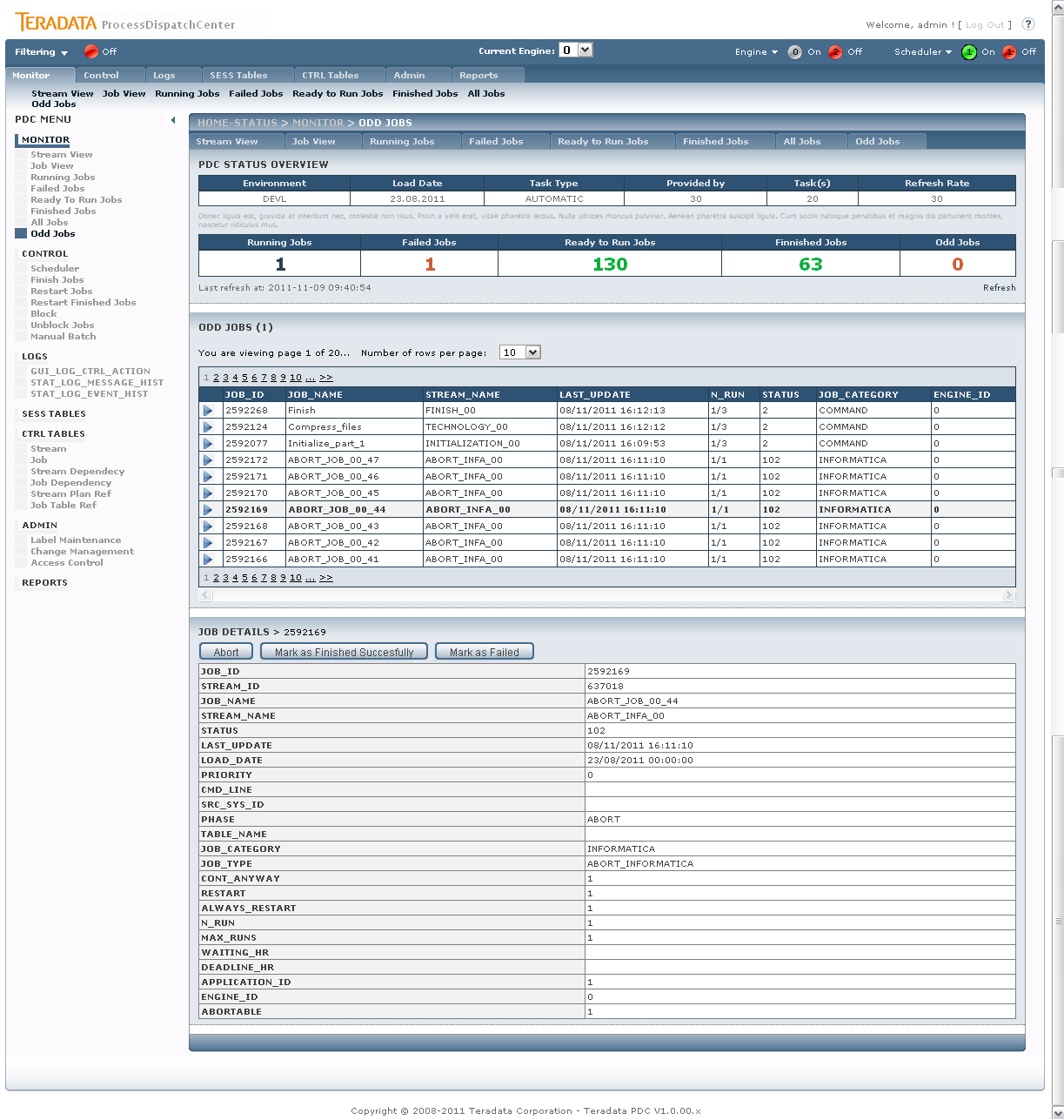


Picture shows Monitoring – All Jobs with details

When jobs in different statuses are displayed, there is no button available on this page.

## Monitor – Odd Jobs

There are two possibilities how to reach this page. The first is using Odd Jobs entry in menu, the second one is clicking on Number of Odd Jobs in Monitoring Header section.



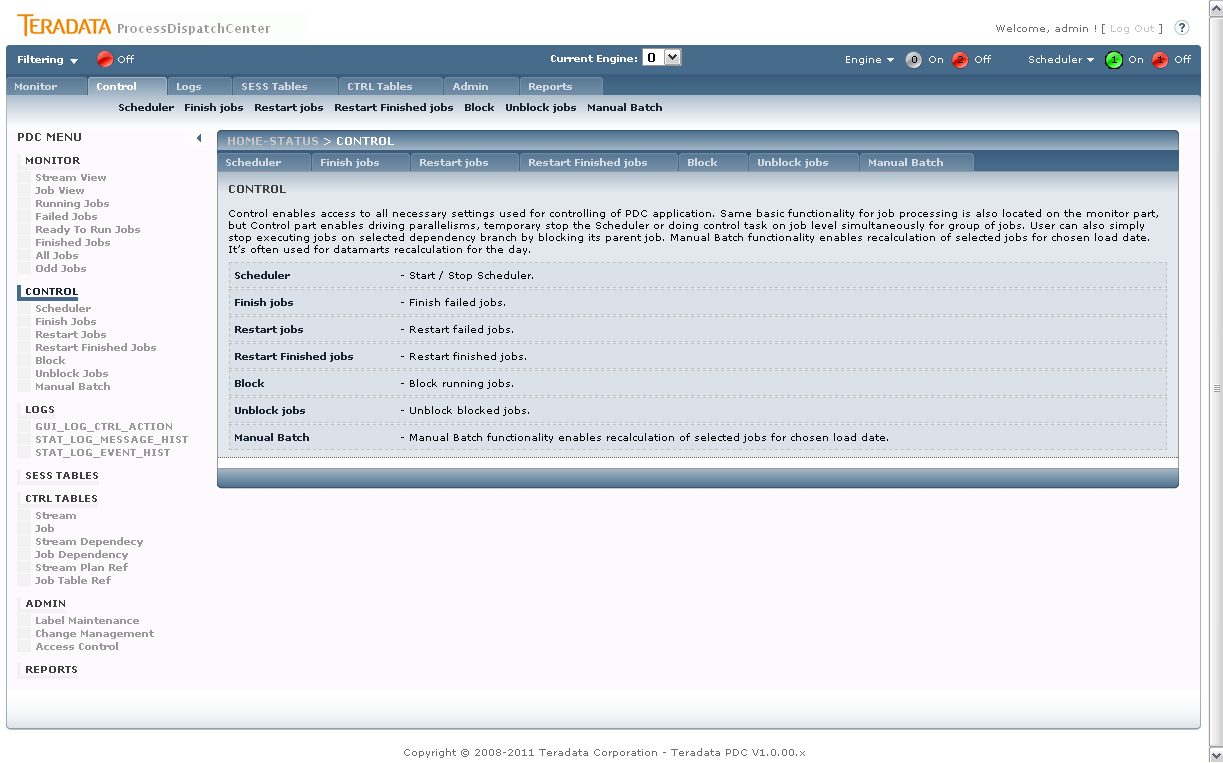
Picture shows Monitoring – Odd Jobs with details

There are shown currently running jobs on this page, which average running time, coming from statistics, was exceeded. This situation gives a signal to supervisor for checking the correctness of job processing. On the top part of job’s detail there is button for job operation. Button has this functionality:

* Abort job – only when job is abort-able is displayed. Clicking on this button, the special job, which aborts job, is started.
* Mark as Finished Successfully – marked jobs as successfully finished. This functionality is used only in case when job finished successfully but job’s metadata from oddly reason aren’t updated.
* Mark as Failed – marked jobs as failed. This functionality is used only in case when job failed but job’s metadata from oddly reason aren’t updated.

# Control processing

Control enables access to all necessary settings used for controlling of PDC application. Same basic functionality for job processing is also located on the monitor part, but Control part enables driving parallelisms, temporary stop the Scheduler or doing control task on job level simultaneously for group of jobs. User can also simply stop executing jobs on selected dependency branch by blocking its parent job. Manual Batch functionality enables recalculation of selected jobs for chosen load date. It’s often used for datamarts recalculation for the day.

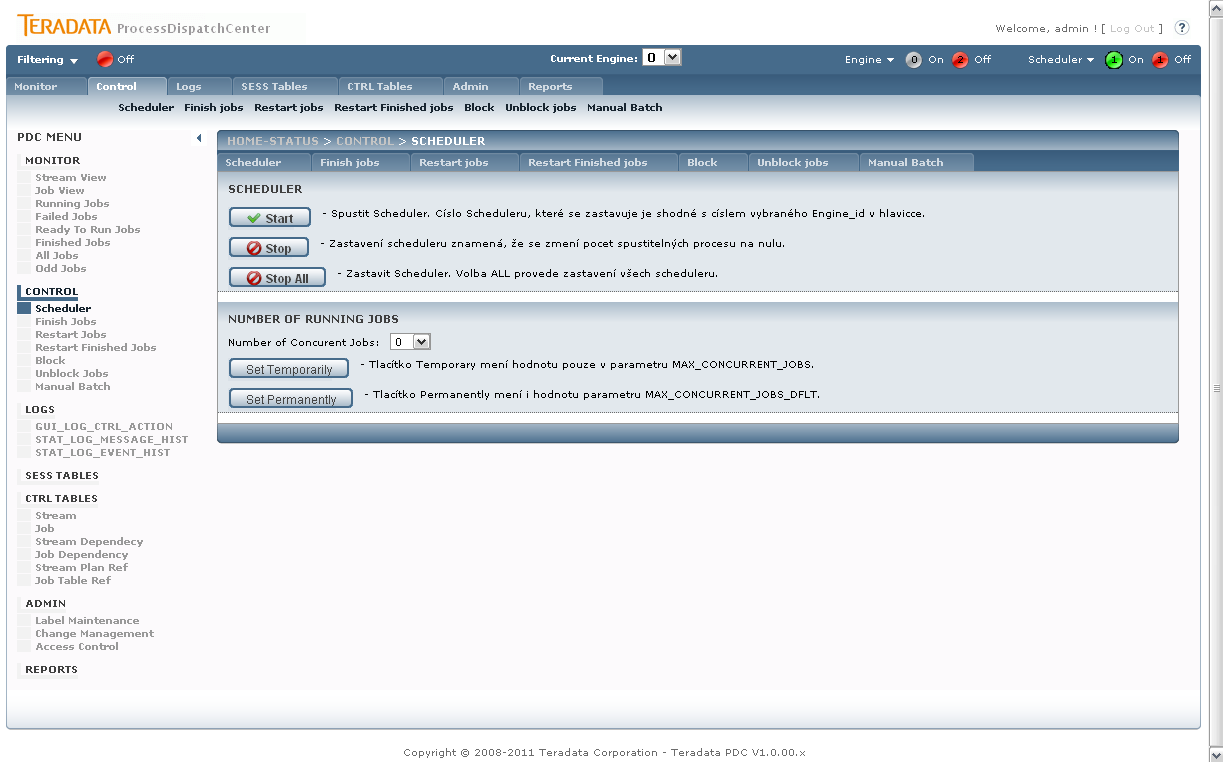


Picture shows the main Control page.

On Control page functionality description is displayed only.

## Control – Scheduler

This page is used for Scheduler control. You can simply stop the Scheduler or change maximal number of currently running jobs for each Scheduler.



Picture shows Control – Scheduler page.

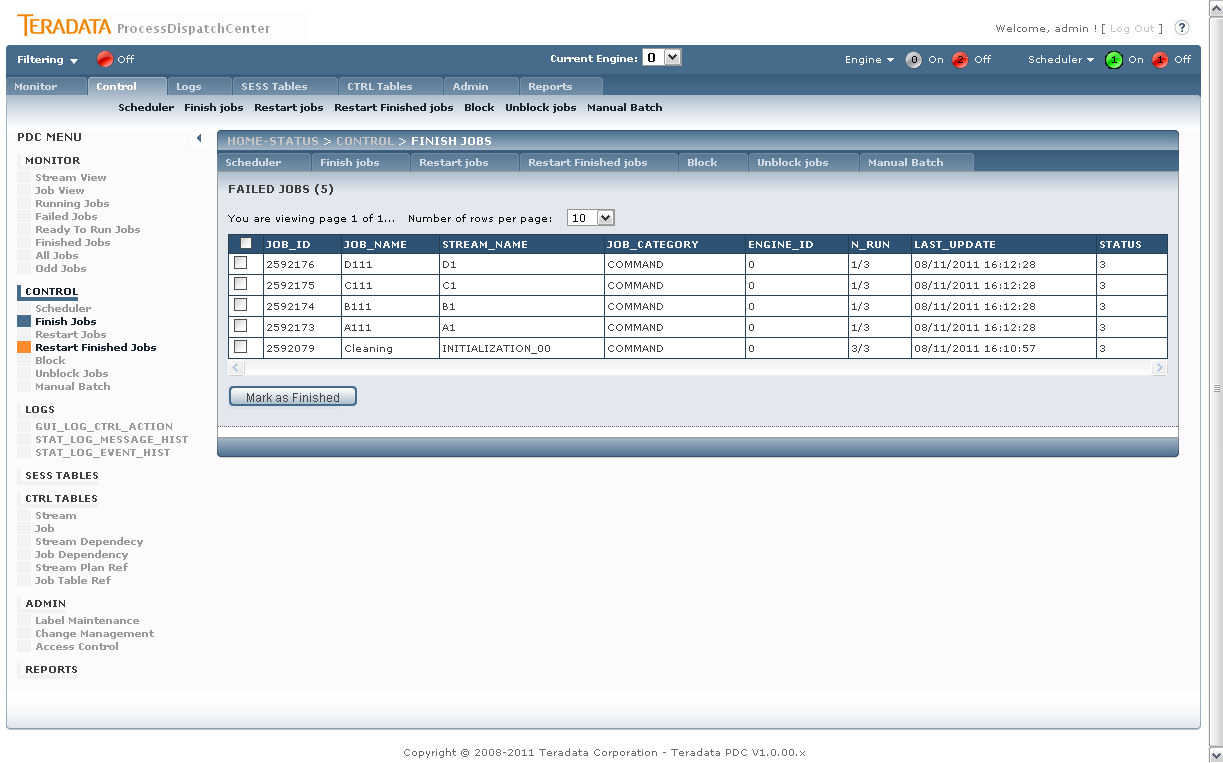
There are several buttons for Scheduler control on this page:

* Start – Start of Scheduler. This action returns Scheduler possibility supply jobs to Engine for launching. The engine\_id number of Scheduler is taken from Status Bar value of Current Engine.
* Stop – Stop of Scheduler. This action removes Scheduler possibility to choose jobs for processing. Factually this action set value of number of concurrent running jobs to zero. No action is applied on already running jobs. The engine\_id number of Scheduler is taken from Status Bar value of Current Engine.
* Stop All – stops all Schedulers (for all Engine\_ids).
* Number of Concurrent Jobs – you can set the maximal number of concurrently running jobs. Please remember that there are also limits on job category and subcategory which are real time effective.
* Set Temporarily – set the number of concurrent jobs temporarily, it means for current job processing. Next day initialization returns the value of concurrently running jobs to default. The engine\_id number of Scheduler is taken from Status Bar value of Current Engine.
* Set Permanently - set the number of concurrent jobs permanently, it means that also default is changed. The engine\_id number of Scheduler is taken from Status Bar value of Current Engine.

## Control – Finish Jobs

This page is used for ending of failed jobs; in this case it means jobs having status Failed. The functionality is the same as it is when button Mark as Finished on Monitor – Failed Jobs is used, but there are some ads on this page:

* Multiple or all jobs can be selected
* Also jobs with not already consumed limit of restarts can be selected

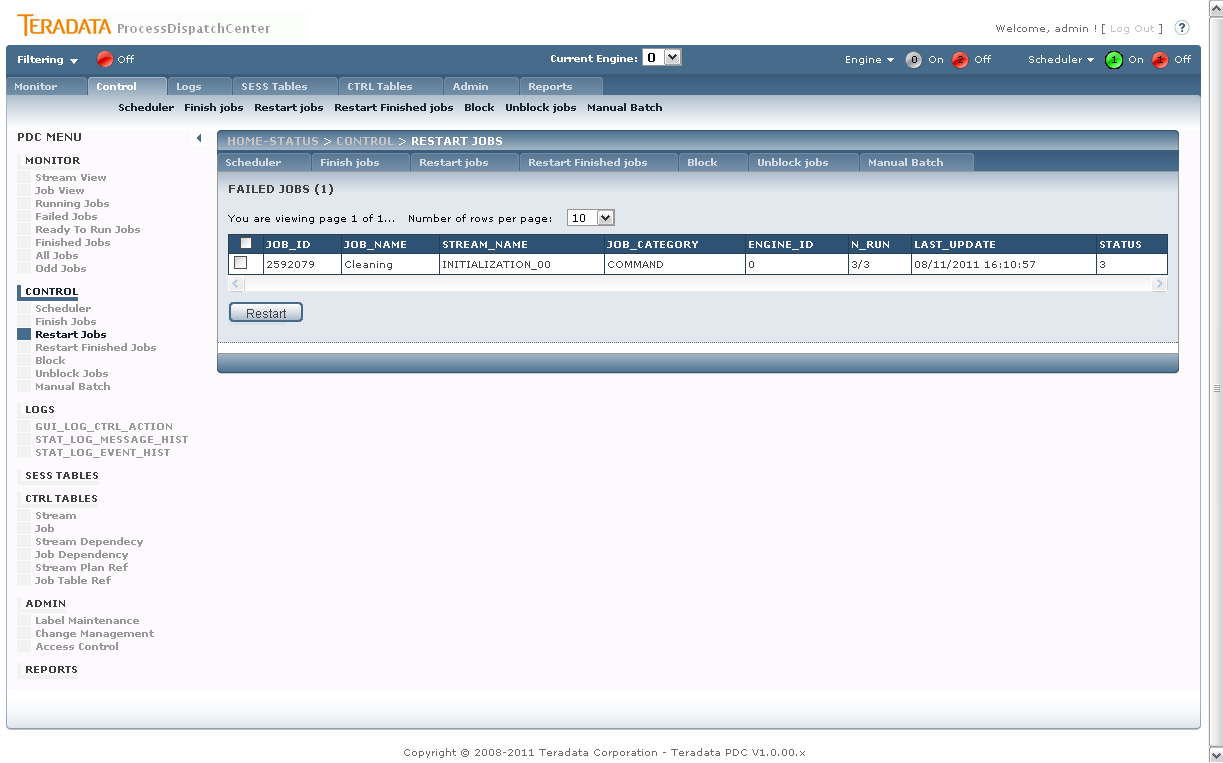


Picture shows the Control - Finish Jobs page.

Pressing the Mark as Finished button the signal Finished is send to the jobs which causes ending of job immediately.

## Control – Restart Jobs

This page is used for restarting of failed jobs, in this case it means jobs not only having status Failed, but also jobs already consumed all possibility for auto restart by Scheduler.



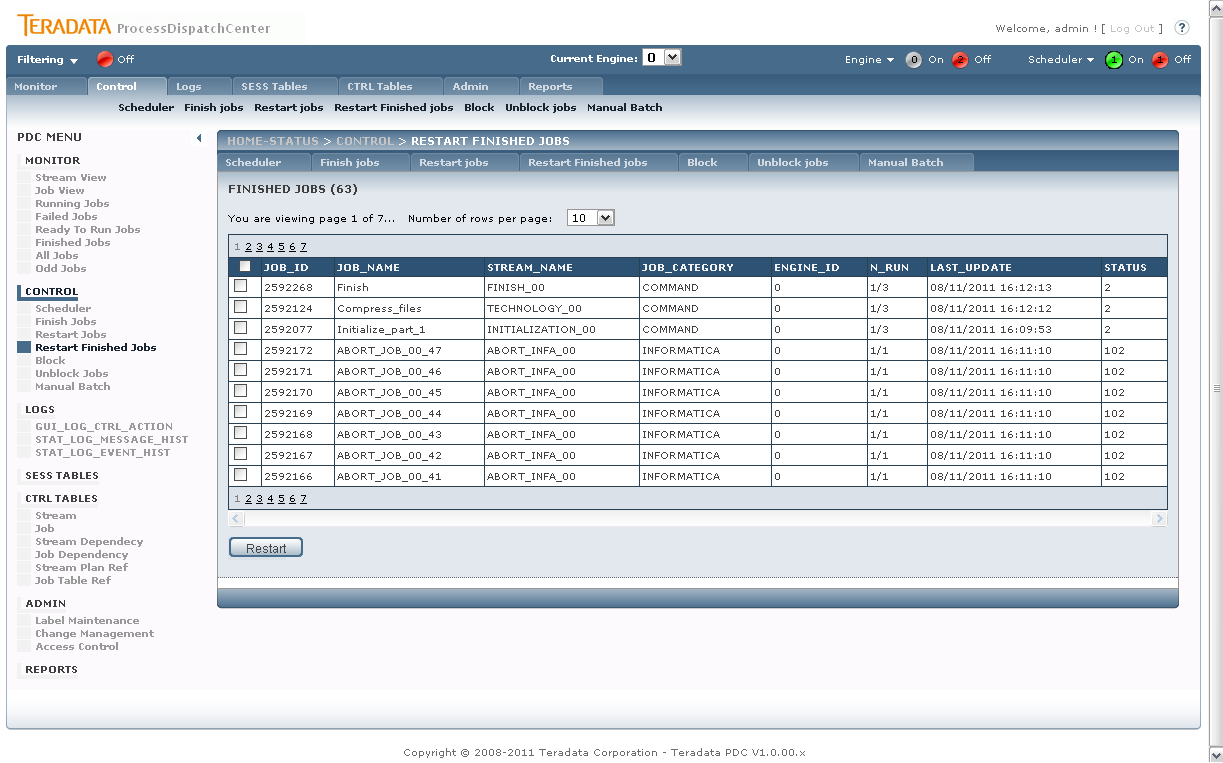
Picture shows the Control - Restart Jobs page.

On the page you can select multiple or all jobs for processing. The action executed on jobs depends of value of always\_restart parameter and button which is pressed:

* Restart – job is returned to Scheduler by increasing value of max\_runs parameter by the value of max\_runs parameter from CTRL\_JOB table. The value of restart parameter is set to true even if resume of job is permited.
* Resume – job is returned to Scheduler by increasing value of max\_runs parameter by the value of max\_runs parameter from CTRL\_JOB table. The value of restart parameter is set to false only when value of always\_restart is set to false also. It means that resume has the same functionality as restart for jobs with value of always\_restart parameter set to true.

## Control – Restart Finished Jobs

This page is used for restarting already finished jobs.

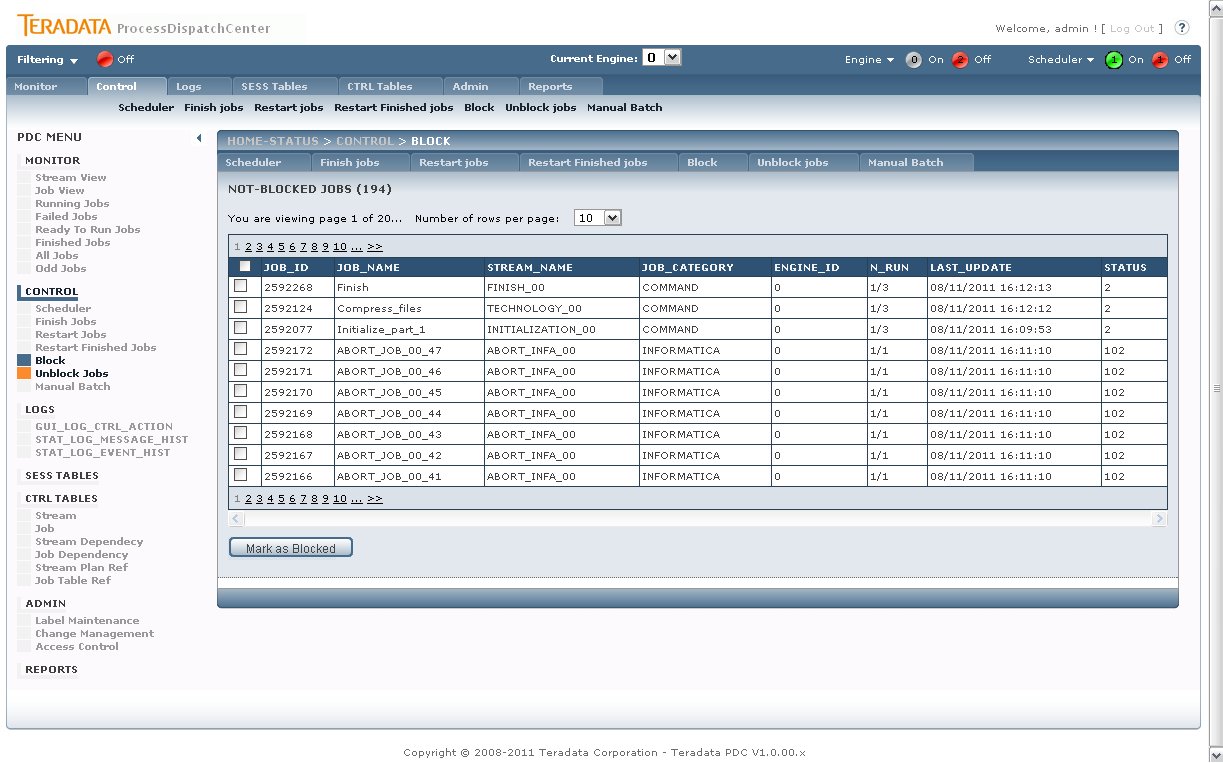


Picture shows the Control - Restart Finished Jobs page.

Please note that even if multiple or all jobs can be selected, the jobs for restart has to be independent each other in case of job to job dependency therefore Scheduler doesn’t know anything about job’s dependency at this time.

## Control – Block Jobs

This page is used for blocking jobs.

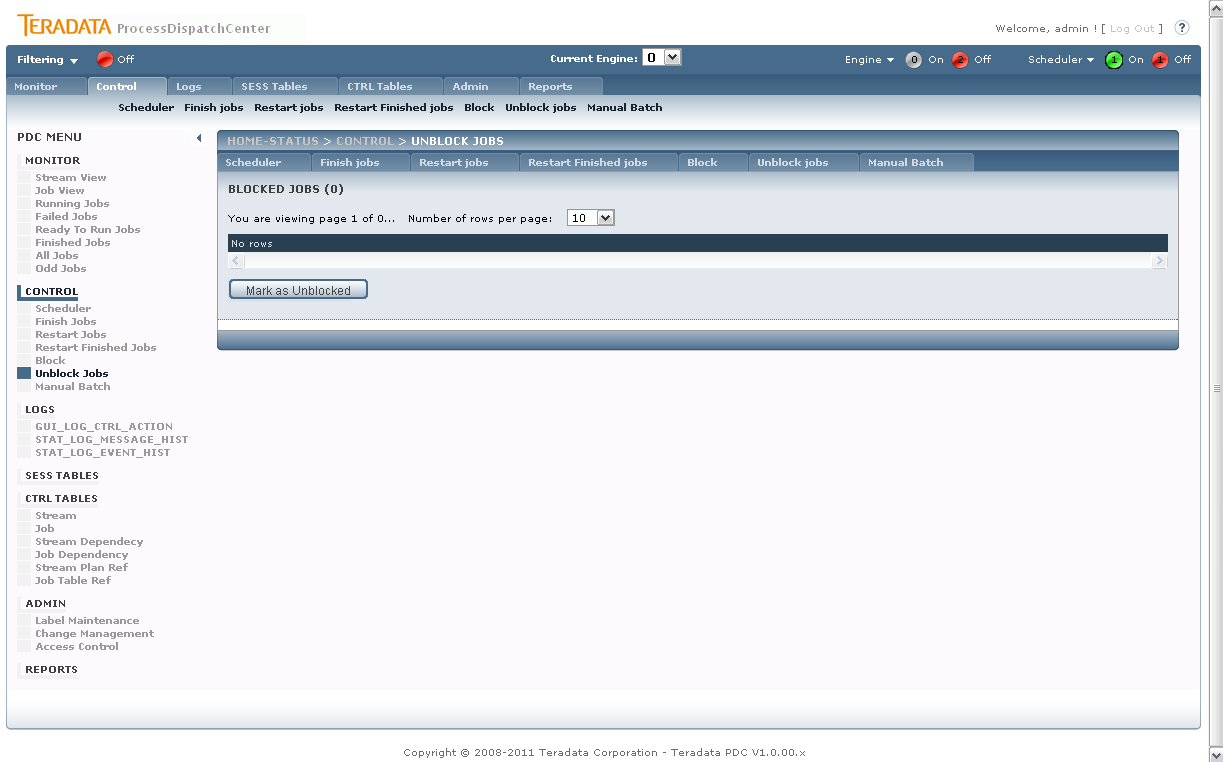


Picture shows the Control - Block Jobs page.

Blocking functionality is used for temporarily removing some jobs from processing. It can be useful if error in processing is found and supervisor wants to protect wrong data to be processed by next jobs. In practice all not already blocked jobs can be blocked, but blocking has meaningful effect only on jobs not started yet. Therefore when finished or running job is selected for blocking, the not launched children has to be found for blocking instead of selected job.

## Control – Unblock Jobs

This page is used for unblocking jobs.



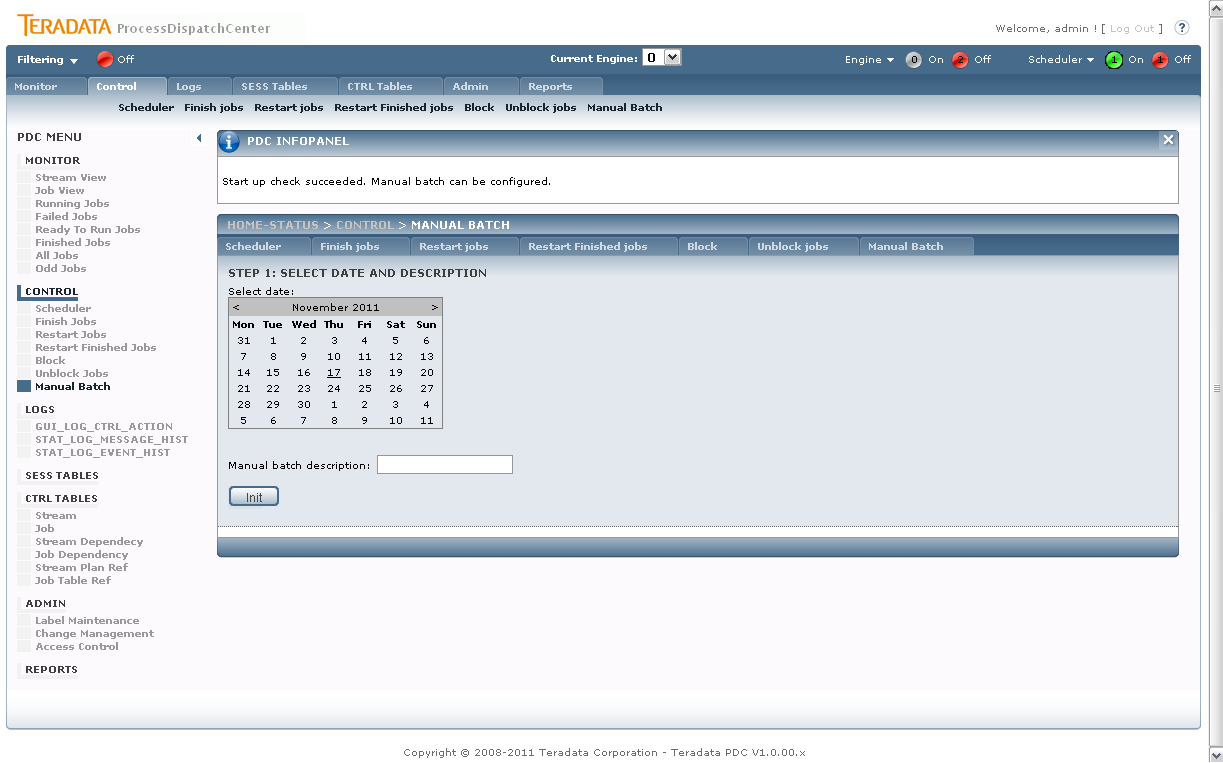
Picture shows the Control - Unblock Jobs page.

Only blocked jobs are shown on page. Multiple or all jobs can be selected. Unblocking returns jobs its original status value.

## Control – Manual Batch

This page is used restarting group of jobs respecting its dependency. Load date for processing also can be chosen.

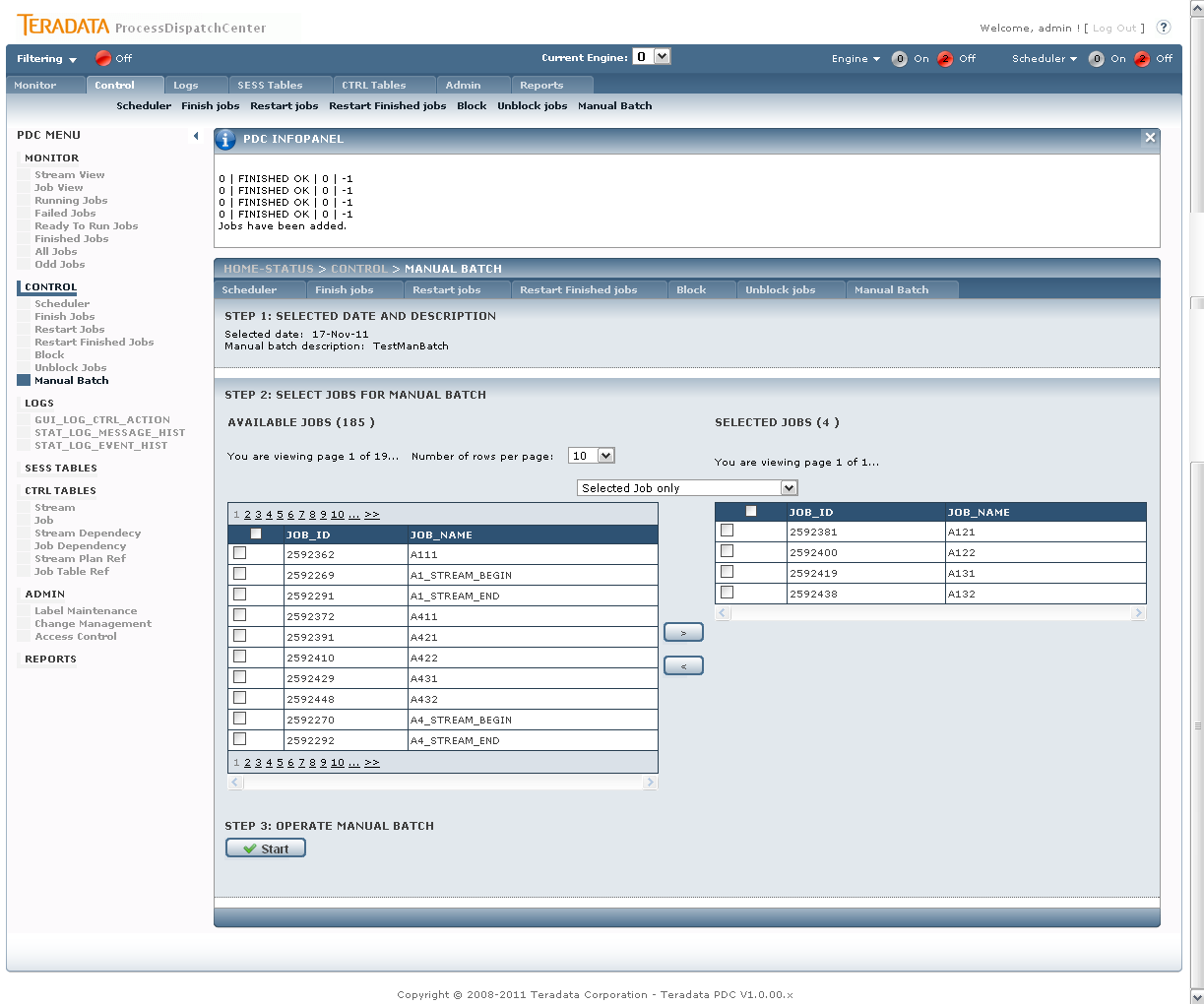
#### Step 1 – Select Date and Description



Picture shows the Control – Manual Batch – Select Date and Description page.

PDC application has implemented several checks if initialization of Manual Batch is possible or not, therefore initialization of Manual Batch during standard or other Manual Batch processing can cause unexpected job’s processing. If Manual Batch initialization is allowed, the page for load date and description appears.

#### Step 2 – Select Jobs for Manual Batch



Picture shows the Control – Manual Batch – Select Jobs for Manual Batch page.

On the page two tables are displayed. The left table contains jobs not selected for Manual Batch processing; the right one contains already selected jobs. There are four possibility how to get job from left table to the right one:

* Selected Job Only – the simplest possibility, you select job(s) in left table ane move them into right one. This option also works for reverse selection from right table to left one.
* Selected Job and Children – this option selects selected jobs and all its children. This option is useful when we have datafile and would like to select all jobs which process this datafile.
* Selected Job and Parents - this option selects selected job and all its parents. This option is useful when we want to refill some table and would like to select all jobs which prepared data for it.
* Selected Job and Child and Parents – union of the second and the third option.

#### Step 3 – Operate Manual Batch

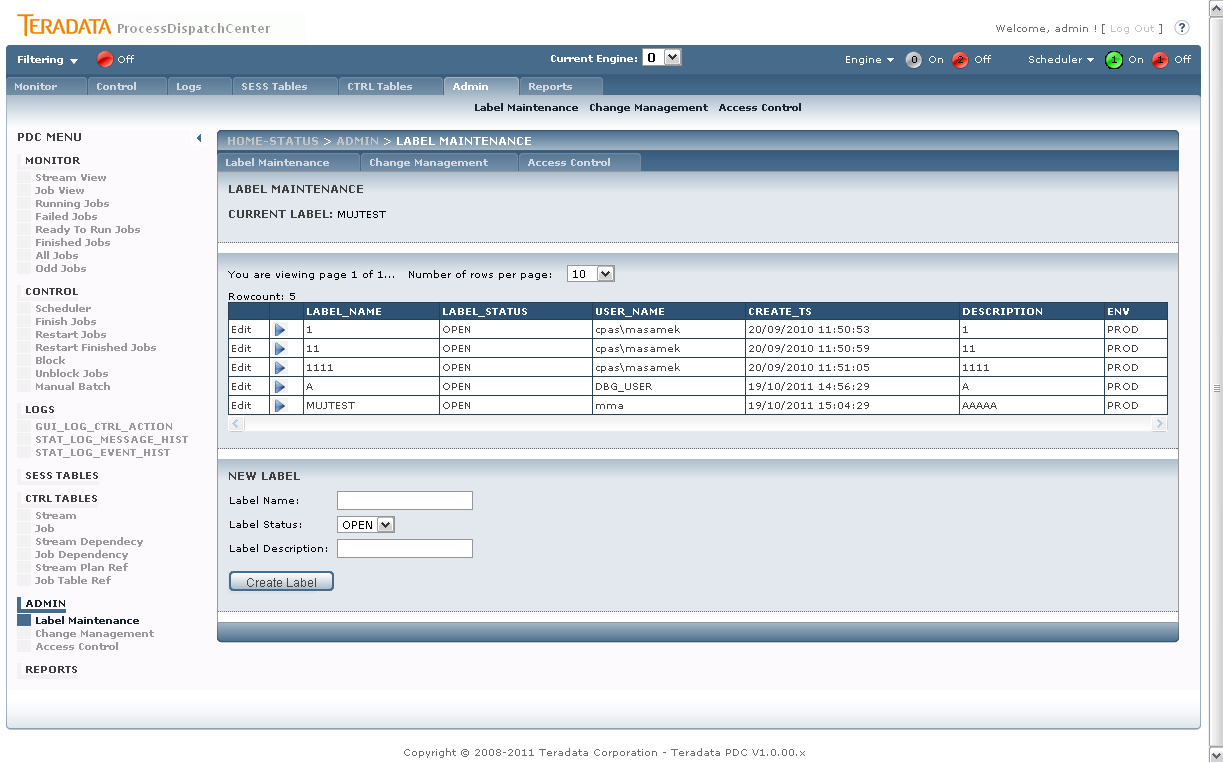
There are two buttons available in this step. Pressing Start button causes Manual Batch processing contrary Cancel option finished Manual Batch without processing.

# Metadata management

Admin section contains necessary administrative tasks such as label maintenance, change management process, access control and so on.

## Admin – Label maintenance

Labels are used for identification of metadata changes which is necessary for correctness of change management process.

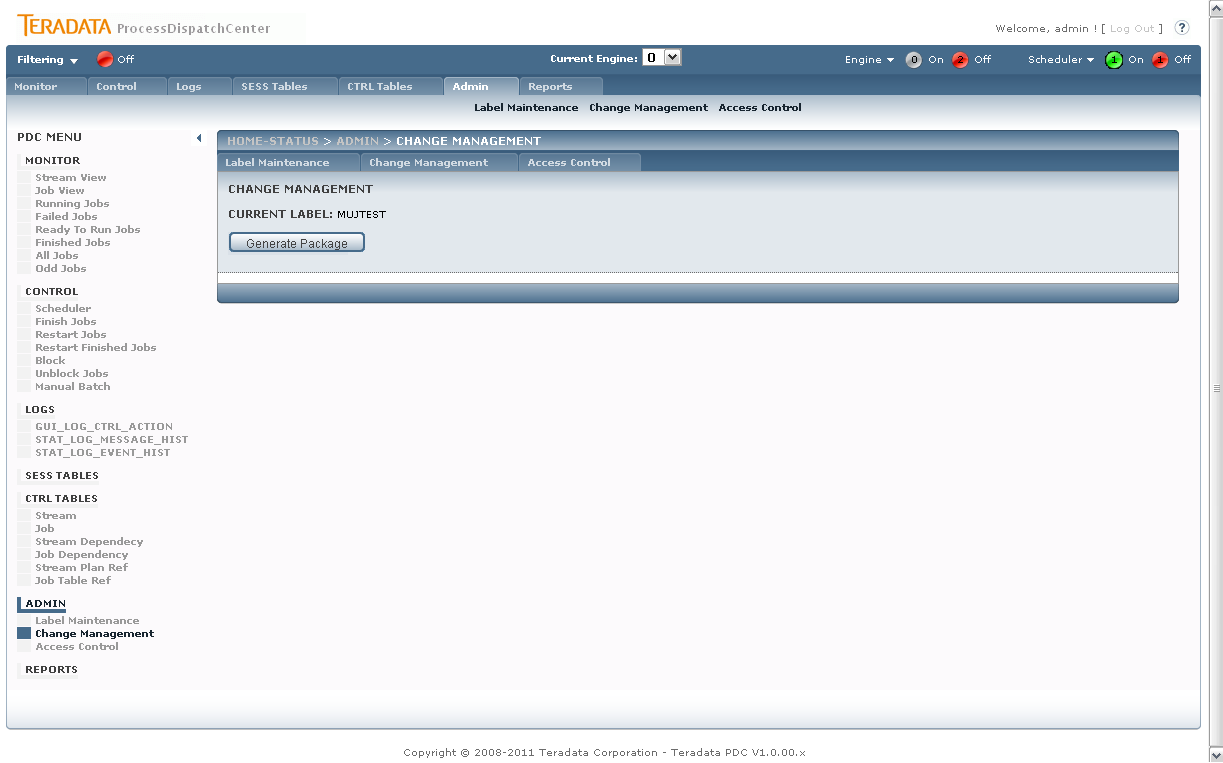


Picture shows Admin – Label maintenance page.

In middle section of the page the label selection is permitted. Only opened labels can be chosen. Bottom section can be used for new label creation. Only label name and description can be entered. If some label is already chosen, the label name is displayed in top section.

## Admin – Change Management

Change management creates SQL file with all changes made in CTRL tables labeled with chosen label. This file is used for tracking these changes in test and production environment.



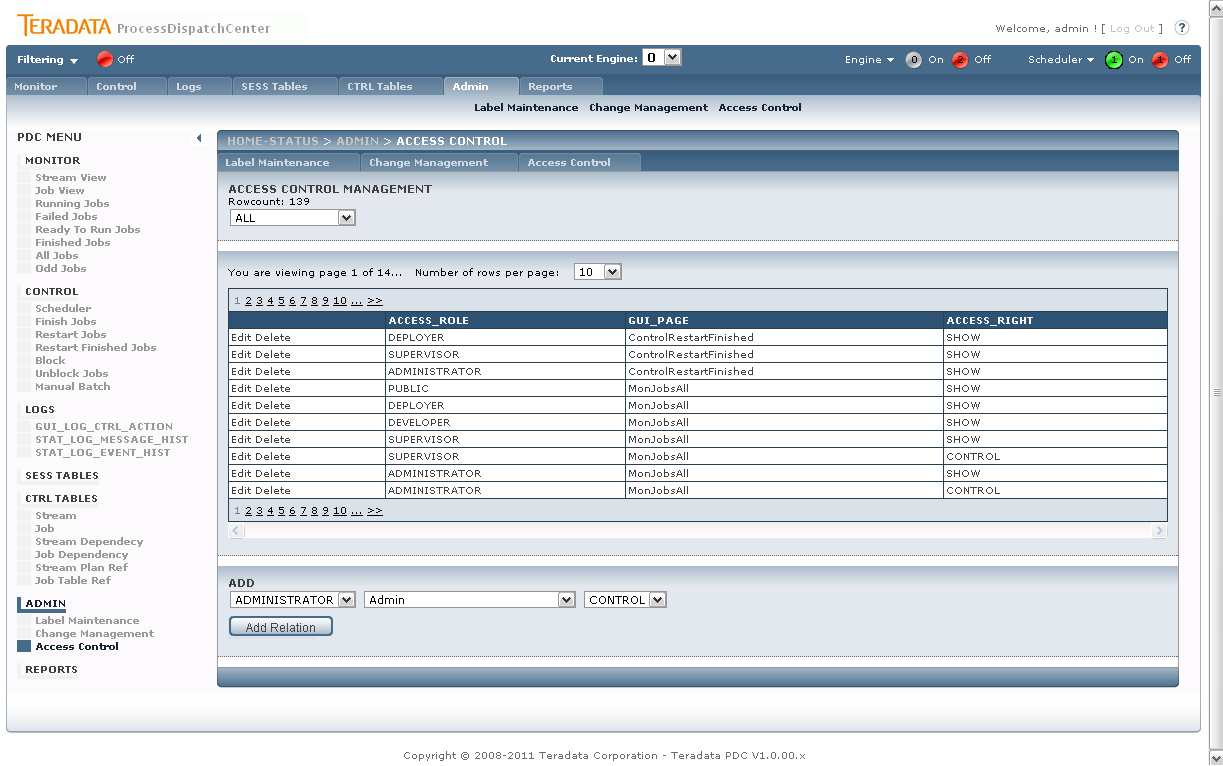
Picture shows Admin – Change Management page.

When pushing Generate Package button, process will ask you for file name for script save.

## Admin – Access Control

Access Control page is used for access rights granting. Generally there are two types of rights related to page:

* SHOW – user can see page content
* CONTROL – function buttons are active



Picture shows Admin – Access Control page.

The rights are not granted on user level, but on group level. Every user with access to PDC application has to be a part of some group.