ConSol
Consulting & Solutions

DOCUMENT

Release Notes ConSol*CM
Version 6.9.4

Author: Michael Siebenborn
Phone: +49 (0) 89 / 45841-100
Mail: michael.siebenborn@consol.de
Version: 1.11
Date: 28.08.2017
Status: Released

ConSol* Software GmbH
Franziskanerstraße 38
D-81669 München
Tel.: +49 (0)89-458 41-100
Fax: +49 (0)89-458 41-111
E-Mail: info@consol.de
Internet: www.consol.de

Geschäftsführer:
Dr. Ulrich Schwanengel
Andrea Stellwag
Bernd Wiserner
Michael Beutner

Handelsregister:
Registergericht München
HRB 97371
Ust-IdNr.: DE 129318612

Table of contents

General update and installation instructions

For an update of ConSol*CM from one version to another two possible ways exist:

- Distribution installation
  The distribution is installed into the application server. For an update every local configuration, like the data source configuration, has to be saved before and reconfigured afterwards.
  This type of update ensures that really every change between the versions is installed.
  This type of update is recommended for updates of the major or minor version, e.g. for an update from 6.6.3 to 6.7.5.
- EAR / WAR Update
  For this type of update of the ConSol*CM, the EAR (cm6.ear, cmrf.ear) and WAR (cm-track.war) files of the new version have to be installed into the application server. Additionally every installation related changes described in the chapters 'Update and installation instructions' have to be applied manually. The changes have to be applied for every version between your original CM version and the new CM version, e.g. for an update from 6.6.3 to 6.6.7 the instructions of the versions 6.6.5, 6.6.6 and 6.6.7 have to be checked.
  This type of update is only recommended for updates within a minor version.

Additionally for every type of update, the 'Update and installations instructions' chapter has to be checked for further important notes.

If available, the solution specific Release Notes have to be checked too.

1 Version 6.9.4.0 (01.12.2014)

Version 6.9.4.0 includes 6.9.3 versions up to 6.9.3.5, 6.9.2 versions up to 6.9.2.10, 6.8 versions up to 6.8.5.6 and 6.7 versions up to 6.7.13

1.1 Update and installation instructions

Boolean fields represented as checkboxes show a different default behavior now (see section 1.3.1) allowing the value *null* after saving while setting *false* previously. This may require adjustments of scripts using the value *false* for unchecked previously. Please see section 4.2.2 for a way how to deal with this and a script helping to identify usages the affected fields.

1.1.1 Data warehouse JMS communication channel not supported anymore

Please note that the JMS data warehouse communication channel is not supported anymore. Only DIRECT mode is available. Please adjust configurations accordingly. See section 7.1.2 for more details.

1.1.2 NIMH New Incoming Mail Handler (#620992)

The NIMH new incoming mail handler can replace processing the incoming e-mail by MULE. Currently both are supported. In future releases of CM6 incoming mail processing by MULE may be removed.

The configuration properties for the NIMH will be added automatically during the update of an existing system.

The capabilities of NIMH in a clustered environment are no different from MULE for this release. This means you can configure and run it only on one node in a cluster. All other nodes, not running NIMH, also must have the ESB service disabled for proper operation.

Switching from MULE to NIMH on production systems should be prepared by adjusting the NIMH configuration including applying all necessary changes to the incoming mail scripts. The next step is shutting down MULE and making sure it finishes processing e-mails, reviewing backuped mails and reprocessing or deleting these. After that NIMH can be enabled and started. It should start processing the incoming e-mails then.

On development and test systems the actual switch to NIMH can be done at any time after configuring NIMH just by setting new dedicated configuration property **`cmas-core-server.nimh.enabled`**`=true`. The switch is done automatically during runtime, there is no need for a server restart. This also allows for switching back to MULE, if desired.

When you switch from MULE incoming mail processing to the NIMH, please make sure that MULE has processed all mails before making the switch. Otherwise some incoming mails may not be processed at all and thus may not be visible in the system.

Please be aware that when switching the previous mails backed up as “unparseable” for example will not be visible in the Admin-Tool any more. Since the storage of these mails differs between MULE and NIMH these can only be shown for the current mail processor. They are not deleted and will show up again when switching back to the original mail processor. So the

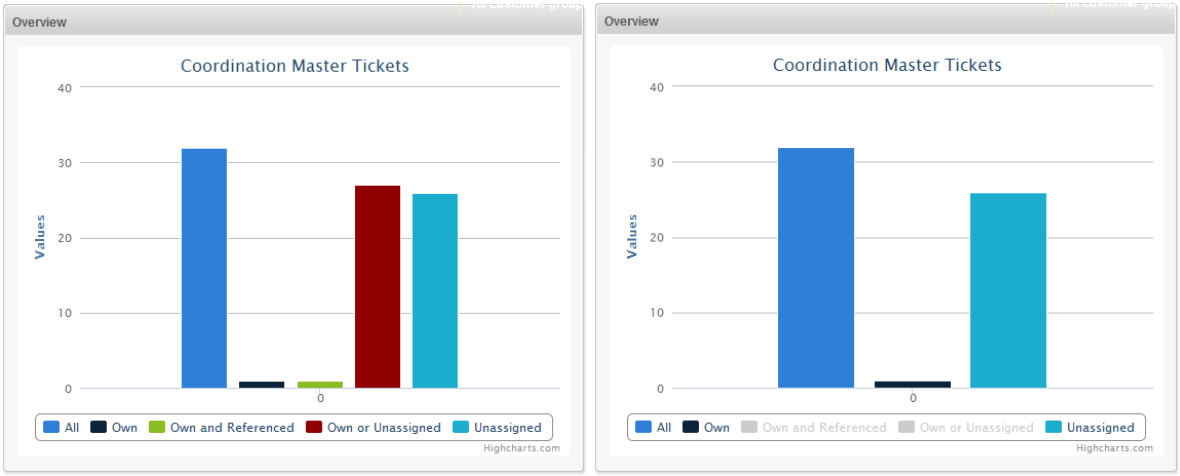
Admin-Tool shows the backuped mails for NIMH when NIMH is active and for MULE when MULE is active, but not all backuped mails together.

Please refer to section 1.3.2 for further details on configuration and use of this functionality.

1.2 New Features

1.2.1 Dashboard on the Web Client Overview Page (#624856)

This release introduces the dashboard on the overview page. This functionality allows to display graphical charts or table representations of data in the system. An example is the ticket count of the ticket list by filter.
The dashboard generally is disabled after updating an existing system. It will be enabled on a new installation.
In case it is enabled, it is shown on the overview page right after the user logged in. It can be displayed any time by clicking the overview button in the main menu. The dashboard layout, its elements and the data displayed can be configured by the administrator. There are no user preferences to personalize the dashboard structure or content permanently.
The dashboard is interactive meaning its current content values can for example adjust to the view selected in the ticket list. Categories can be hidden when clicking on the element in the legend, please compare the screenshots below.

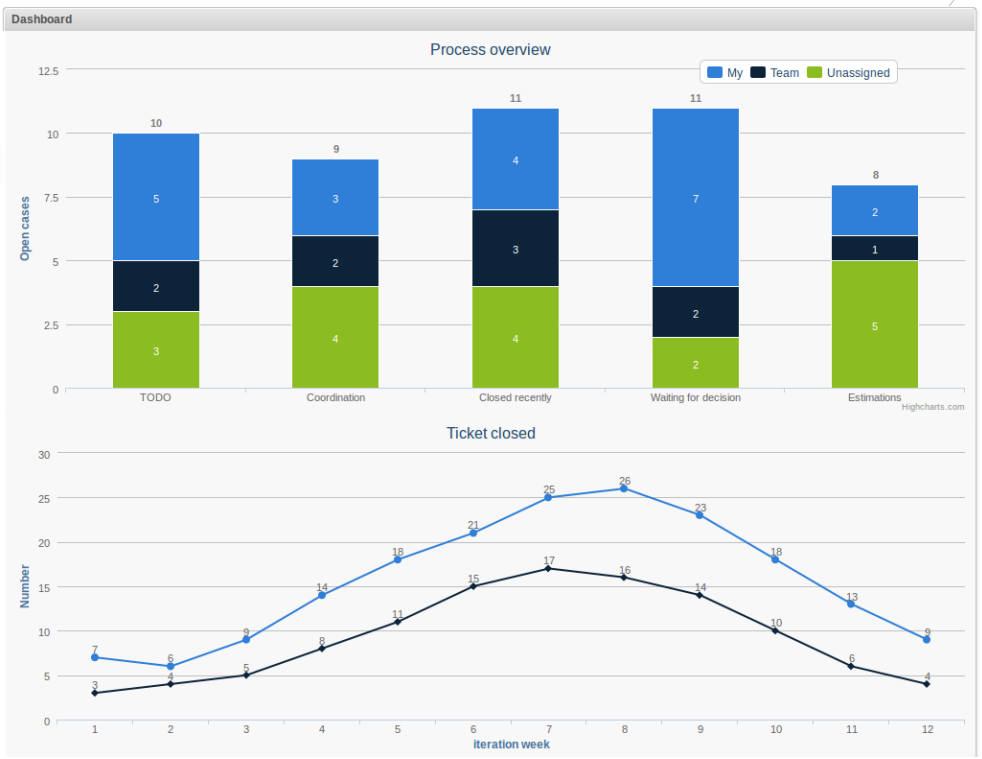


Please be aware that the queries to gather the data to display put additional load on the system! The more complex or general the requested information is, the higher the impact can be! Please note that the dashboard can be disabled as a whole in case the additional performance requirements influence the overall system responsiveness. See below in the section “Disabling the Dashboard and Widgets” for details on how to disable the dashboard.

Example Layout Designs

The different dashboard elements are called widgets. They can be arranged in a table grid. Charts and data tables can be used side by side and several charts and data tables can be used in complex layout. For the charts there are numerous design options to choose from. See the screenshots below for illustration.

Two Chart Widgets in one Column:



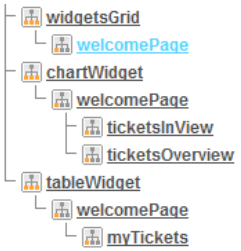
A Chart and a Table widget in one Column:

#	Title	Duration	% Complete	Start	Finish	Effort Driven
0	Task 0	5 days		01/01/2009	01/05/2009	✔
1	Task 1	5 days		01/01/2009	01/05/2009	
2	Task 2	5 days		01/01/2009	01/05/2009	
3	Task 3	5 days		01/01/2009	01/05/2009	
4	Task 4	5 days		01/01/2009	01/05/2009	
5	Task 5	5 days		01/01/2009	01/05/2009	✔
6	Task 6	5 days		01/01/2009	01/05/2009	
7	Task 7	5 days		01/01/2009	01/05/2009	
8	Task 8	5 days		01/01/2009	01/05/2009	
9	Task 9	5 days		01/01/2009	01/05/2009	
10	Task 10	5 days		01/01/2009	01/05/2009	✔

Widget Creation and Layout

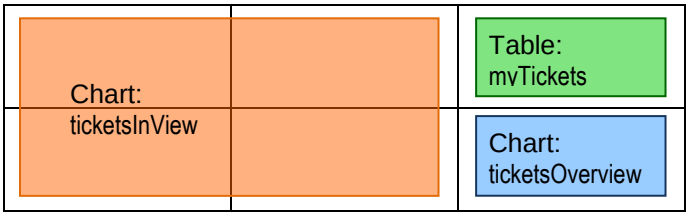
The dashboard generally is configured in the page customization for the CM6 overview page and its subcategories. Layout and Widget creation is done using the customization *widgetsGrid/welcomePage*. This scope has one attribute called *layout*.

A new widget will be added to the page customization tree automatically when it is added in the value of this layout attribute. After saving and reloading the page it is available in the tree for further configuration.



The layout is defined by the complex value of the layout attribute. The grid configuration uses the configuration pattern applied in the CM/Track product also. The general syntax rules are:

- Each row is represented as an array of elements: `[x,y,z]`
- The grid starts with the upper-left corner `(0,0)` and it is built up row after row.
- `null` is a reserved key word for an empty cell.
- A widget is described by its name and its type, separated by a colon, i.e '`ticketsInView:Chart`'.
- The name for a specific widget must be unique.
- The type can currently be either `Chart` or `Table`. The type is recognized only once. The type of a widget can be omitted, if it is repeated multiple times: `[ticketsInView:Chart, ticketsInView, ticketsInView]`
- The widget can occupy multiple adjacent rows and columns: `[ticketsInView:Chart, ticketsInView, myTickets:Table], [ticketsInView, ticketsInView, newTickets:Table]` defines that the widget `ticketsInView` uses two rows and two columns from the upper left corner in a two row, three column grid. Schematically this layout looks like this:



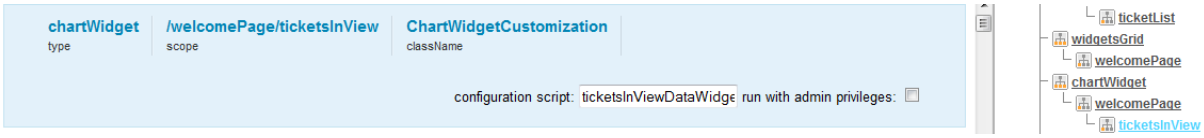
Disabling the Dashboard and Widgets

The dashboard can be completely disabled in removing the value from the attribute *layout* in the section *widgetsGrid/welcomePage*. Without a layout configuration no widgets will be shown and the widget scripts will not be executed. The overview page will have the appearance from earlier CM6 releases.

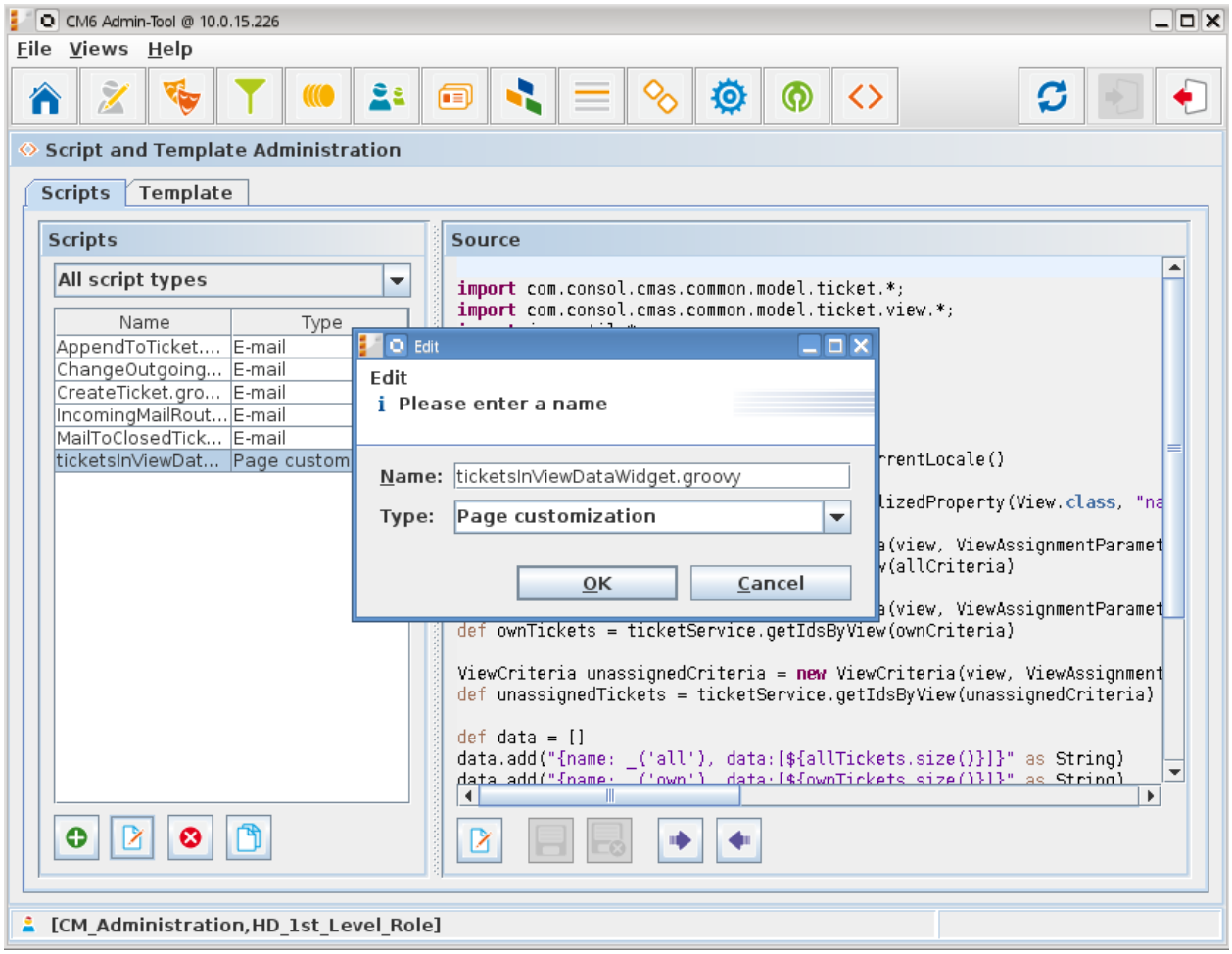
Alternatively, the data script association (for further details see below in the section “Widget Data and Configuration Sourcing”) can be disabled for each widget in order to disable the dashboard. Changing the widget configuration script value to an empty value disables the respective widget. In a single widget layout this also disables the dashboard as a whole.

Widget Data and Configuration Sourcing

A widget referenced in the layout attribute value will show in the page customization tree. The location in the tree depends on its type: *chartWidget/welcomePage/<widgetName>* for charts and *tableWidget/welcomePage/<widgetName>* for tables.



The header of the customization provides an entry field which can be used to enter the name of a Groovy script. The script name identified here must be a script that has been created in the Admin-Tool script administration using the script type “page customization”.



Please note: Faulty scripts can make the web client unusable when referencing them in the page customization. In order to be able to deal with this the script source code can be removed or commented out in the Admin-Tool.

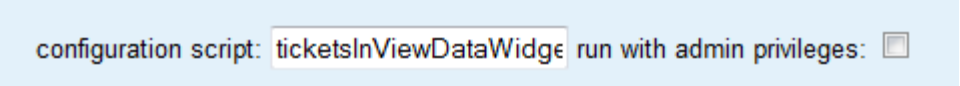
The execution of this groovy script is a core part of the customization. The script overwrites the configuration data provided in the page customization. The script must return a map of variables which correspond to the defined widget properties. Please note that the value returned by the script always replaces the one defined in the customization property. The values are *not* merged! The script thus will override any widget property value set in the customization, so make sure that the desired property is not set inside the script, if you want to use the customization for property setting.

As a working example the default script is shown here. This script will be added to your system for reference even in an update when the dashboard is disabled. You can find it in the Admin-Tool script administration by the name “ticketsInViewDataWidget.groovy” with the type “Page customization”.

```
import com.consol.cmas.common.model.ticket.*;
import com.consol.cmas.common.model.ticket.view.*;
import java.util.*;
import java.util.Map.Entry;

if (viewId == -1) {
  return [visible: 'false']
}
def engineerLocale = engineerService.getCurrentLocale()
def view = viewService.getById(viewId)
def viewName = localizationService.getLocalizedProperty(View.class,
        "name", viewId, engineerLocale)

ViewCriteria allCriteria = new ViewCriteria(view,
        ViewAssignmentParameter.allTickets(),
        ViewGroupParameter.allTickets(),
        new ViewOrderParameter())
def allTickets = ticketService.getCountForView(allCriteria)

ViewCriteria ownCriteria = new ViewCriteria(view,
        ViewAssignmentParameter.allTickets(engineerService.getCurrent()),
        ViewGroupParameter.onlyOwnTickets(),
        new ViewOrderParameter())
def ownTickets = ticketService.getCountForView(ownCriteria)

ViewCriteria unassignedCriteria = new ViewCriteria(view,
        ViewAssignmentParameter.allUnassignedTickets(),
        ViewGroupParameter.onlyUnassignedTickets(),
        new ViewOrderParameter())
def unassignedTickets = ticketService.getCountForView(unassignedCriteria)

def data = []
data.add("{name: _('all'), data:[${allTickets}]}" as String)
data.add("{name: _('own'), data:[${ownTickets}]}" as String)
data.add("{name: _('unassigned'), data:[${unassignedTickets}]}" as String)


return [series: "[${data.join(',')}]" as String, visible: 'true',
        chart: "{type: 'column'}", title: "{text: '${viewName}'}" as String,
        tooltip:"{headerFormat:''}" ,
        localization: "de: {all:'Alle', own:'Eigene', unassigned:'Unzugewiesene'},"+
```

```
                    "en: {all:'All',  own:'Own', unassigned: 'Unassigned'}"];
```

Scripts executed in context of the user role: Other groovy scripts defined in the Admin-Tool usually are executed with global administrative permissions. This is not generally desirable for the scripts providing the data for the dashboard, so by default the scripts used here run with the privileges of the user logged in. This is important because results from execution of the API methods vary depending on the permissions, for example `ticketService.getAll()` will return only tickets the current user can at least read for the user, and absolutely all tickets in the system when executed as admin.
However, there are cases where from a statistical point of view, the need to report over all data exists. Administrative permissions must be used to achieve this. A checkbox in customization GUI was introduced next to input field for setting up the customization groovy script for global and chart script customizations. When checked the referenced script will be run with administrative privileges to globally select the data. For the usual case of user-specific data selection admin privileges are not required for the script. Please see the screenshot and section 1.3.14 for additional details.



Available Widgets

There are two widgets available for the user:

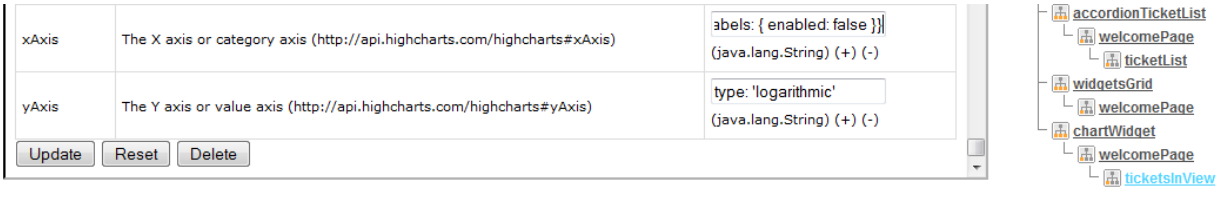
- ChartWidget
- TableWidget

The widgets have all their configuration options made available by means of the page customization. All are available for the script described above as well. The widgets support a set of common features: They allow more than one usage of the same type of component on the page with independent configuration. Additionally the widgets support localized values and visibility manipulation. Besides that each widget has a large set of specific settings.

General Widget Configuration: Localized Values and Visibility

The supported widget properties can be set by the page customization interface for the overview page.

Specific properties are accessible as customization for the individual widget object:



Localization: Some of the widget properties allow the administrator to introduce locale-dependent strings. Each widget has an attribute *localization* in its page customization object to define these strings. The localization strings one can be introduced in its value in a JSON object:

```
String localization = "de: {title:'Thema'}, en: {subject:'Subject'}"
```

The localization value string will be substituted by use of the **_()** function. An example looks like this:

```
String title="text:_('subject')"
```

The substitution is done by the widget component that can identify the locale for which the component is being rendered. If the corresponding key is not found in the localization property, the name of key is used instead as a default value.
Alternatively the localization string values can be defined within the widget script. The same applies to use of the replacement function. Examples for both are shown in the script example above.

Visibility: Each widget's customization has the visible property which is used to control the visibility of the widget. The default value is set to "false". This property is checked in the first place before the widget is rendered. If it is set to false, no further action is taken. This property is general enough to support the most different use cases not even defined yet (view, roles, engineers, etc.). I can easily be set in a widgets' script as in the example script above. Another example script snippet is shown below: the chart should be shown, if view is *service_customer* and the engineer has *consultant* role.

```
view = viewService.getById(view_id) // view_id is passed in context
if (!view.getName().equals("service_customer")) {
  return {"visible": false}
}

def role = roleService.getById('consultant');
def engineer = engineerService.getById(engineer_id);
if (!getRolesForEngineer(engineer).contains(role)) {
  return {"visible": false}
}
```

Chart Widget

The Chart Widget uses the highcharts library to display data to the end user. It encapsulates the library and provides the customization properties to configure the rendering options of the library.

The page customization object of the widget is the location to define the set of properties for the data display. The number of possible options for chart design available in the highcharts library is quite high. For this reason the customization object uses options grouping which is related exactly to highcharts API. The following properties list is supported (see also http://api.highcharts.com/highcharts):

chart, colors, credits, drilldown, exporting, labels, legend, loading, navigation, noData, pane, plotOptions, series, subtitle, title, tooltip, xAxis, yAxis

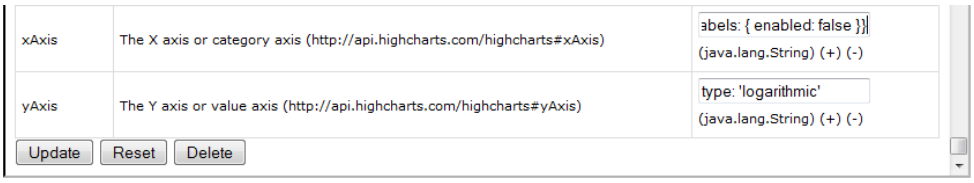
The single option may look similar to the following examples:

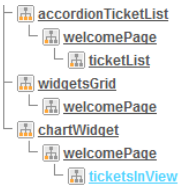
```
chart = "type:'column', pltShadow:false, backgroundColor:'#4dc245', height:
300";"items: [{html:'sometext', style: { left: '100px'; }}]"
```

Note, that the value is a JSON object. For convenience reasons the opening and closing brackets may be omitted:

```
labels = "items: [{html:'sometext', style: { left: '100px'; }}]"
```

The nested JSON notation may appear complex, but it still is easier to manage than introducing a very high number of individual customization's properties. The configuration data is passed to the highchart library directly.
An example shown in the context of the page customization is this:





In the above example a logarithmic scale for the yAxis has been chosen so that small values cannot disappear in comparison with high numbers. However, please be aware that this setting only shows part of the scale without zero for reference, if all values are within a small range.
Please note that the value notation without enclosing in curly brackets is only acceptable for simple values like `type: 'logarithmic'` but not for more complex objects like `{labels: { enabled: false }}`. The latter **must** be a complete JSON object enclosed in curly brackets as a whole. For simple values like the first example the enclosing are omitted.
Any configuration option for a chart widget can be set in the return value of the widget script. Again the example script explained above can serve as an example.

TableWidget

The table widget is the way to present data in grid/table layout in the dashboard. For this end the datatables (http://datatables.net/) library is used. The configuration parameters and data source are configurable via page customization. This is done in the same way as for the chart widget. The properties of the page customization reflect the DataTable API 1.10.0. The list of available properties is:
columns, options, data, localization, features, callbacks

An Example snippet looks like this:

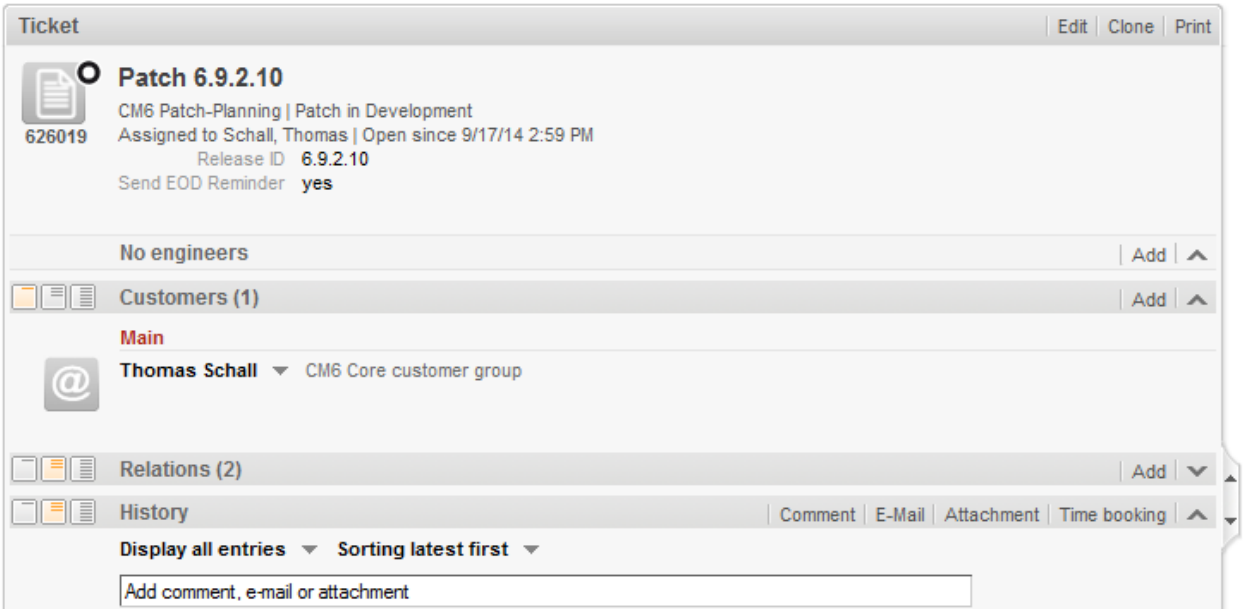
```
columns = [ {id: "title", name: _("Title"), field:"title"}, {id:"duration",
name:_("Duration"), field:"duration"}, {id:"%", name:_("% Complete"),
field:"percentComplete"}, {id:"start", name:_("Start"), field:"start"},
{ id:"finish",  name:_("Finish"), field:"finish"}, {id:"effort-driven",
name:_("Effort Driven"), field:"effortDriven"} ];
```

The above example shows how to create a table widget displaying six columns of data. The widget also uses localized names of the columns in the same way as a chart widget.

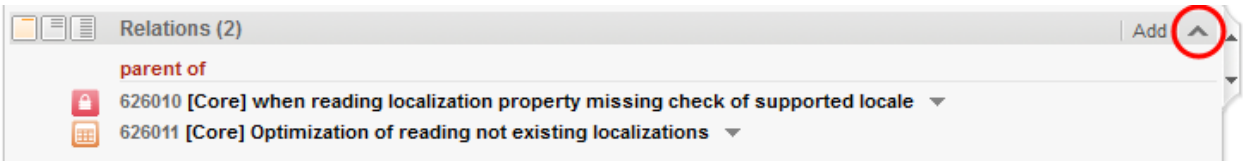
1.2.2 Web Client Section Management: Section Ordering, Collapse and Expand Functionality, Counters (#625340, #625342, #625404)

This new release adds functionality to configure the ordering of page sections and the initial visibility of their content. This applies to the ticket, contact and company pages. Now it is controlled by the administrator, if the content of a section is initially visible when a page is loaded. This differs from before when the user was able to show or hide sections even for newly loaded tickets. Furthermore the order of these sections can now be changed as suitable by the administrator.

In the following example screenshot the engineers section has been moved above the customers section by this configuration. Additionally the engineers and relations sections are displayed initially collapsed, so that their content is hidden. In order to provide initial information the number of entries in this section is shown in its title bar (except for the history).



The display of the content in a collapsed section can be toggled using the arrow icon on the right end of a section's title bar. For a collapsed section this arrow points downwards. Clicking on it will then expand the section contents and the arrow will point upwards, as can be seen in the image below. An expanded section can be collapsed at any time by clicking the arrow icon again.

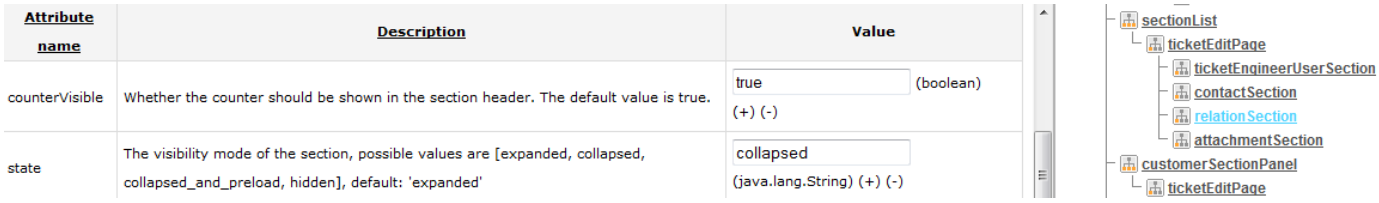


The initial collapsed/expanded state of a section will be set as configured by the administrator. The engineer can change the state for the section on the page, but is not able to set it generally.

This display state as well as the order of the sections can be defined by new page customization parameters.

The state and the visibility of the counter in the section title bar is set by the customizations *state* and *counterVisible* in the *sectionList/ticketEditPage* section entries. The possible values for the state are:

- *expanded* (default, data are shown initially)
- *collapsed* (data are not shown initially and will be loaded only on demand)
- *collapsed_and_preload* (data are not shown initially, but will be loaded)
- *hidden* (the section is completely hidden and cannot be made visible)

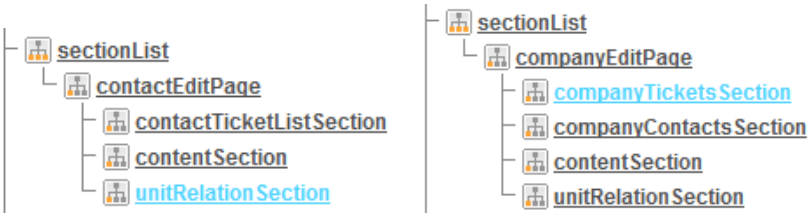


Attribute name	Description	Value
counterVisible	Whether the counter should be shown in the section header. The default value is true.	true (boolean) (+) (-)
state	The visibility mode of the section, possible values are [expanded, collapsed, collapsed_and_preload, hidden], default: 'expanded'	collapsed (java.lang.String) (+) (-)

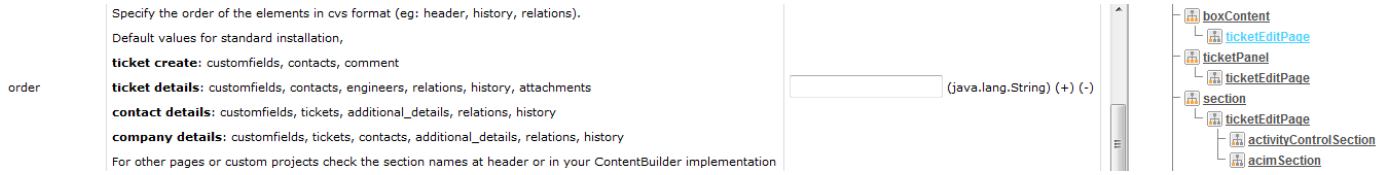
While the value *collapsed* can provide some performance improvement in the initial ticket display for not loading all data, this is not the case when setting the value *collapse_and_preload.*

The possible values for *counterVisible* are the boolean values *true* and *false* with *true* being the default.

These customizations are paralleled on the contact edit page (*sectionList/contactEditPage*) and the company edit page (*sectionList/companyEditPage*) as shown in the screenshots below.



The order of the sections on a page is determined by the list which can be set as the value for the customization attribute *order* in the *boxContent* context of a page. This list is a comma separated value list and its possible element values vary. The available element values and their default order are identified in the attribute description of the page customization for the different pages.



order	Specify the order of the elements in cvs format (eg: header, history, relations). Default values for standard installation, **ticket create**: customfields, contacts, comment **ticket details**: customfields, contacts, engineers, relations, history, attachments **contact details**: customfields, tickets, additional_details, relations, history **company details**: customfields, tickets, contacts, additional_details, relations, history For other pages or custom projects check the section names at header or in your ContentBuilder implementation	(java.lang.String) (+) (-)

The section names actually are listed in the description of the attribute for the respective pages in their default order.

1.2.3 Web Client Label Use in Customer Data (#624463)

The use and configuration of labels for data object group fields for units (contacts and companies) has been changed in this release to better address the following everyday use cases:

- The engineer needs good orientation when working with customer data. For this the fields have be identified easily even when data are already present. This was not guaranteed earlier, so that optimizations in usability were necessary.
- The CM6 administrator needs to create a data object group fields. The expectation is that the description of the newly created field will be used as its label. Creating an extra custom field of type label in not wanted. The previous behavior was not consistent with the creation of custom fields for tickets. On the other hands watermarks should still be supported, if the adminsitrator wants to use this type of labeling.
- Updating a system to the new version should still support the old configuration possibilities (watermarks, no labels). The administrator should be able to manually convert the old configuration to the new one if desired.
- The administrator can define a layout for data object groups without being forced to use fields annotated as labels. Removing this constraint is effectively reducing the number of data object group fields used by half. Administrators need a flexible system for creating layouts.

Please be aware that the changes to provide this new feature changed the default behavior for customer data when creating new data object groups and data object group fields! The layout of existing field groups/fields will be preserved in an update!

The labels for new data object group fields now are rendered in the same way as the ones for ticket custom fields. The presentation of elements created before an update will not be changed. However, both display modes can easily be changed to address the requirements of a specific installation.

These improvements have been implemented by introducing eight new annotations for data object groups and data object group fields.

Annotations for data object groups:

- `layout:show-labels-in-view` (true if not set)
- `layout:show-labels-in-edit` (true if not set)
- `layout:show-watermarks` (false if not set)
- `layout:show-tooltips` (true if not set)

Annotations for data object group fields:

- `layout:show-label-in-view`
- `layout:show-label-in-edit`
- `layout:show-watermark`
- `layout:show-tooltip`

For all existing data object groups the annotation setting `layout:show-labels-in-edit=false` (which will hide standard labels in edit mode) is made during the update to this

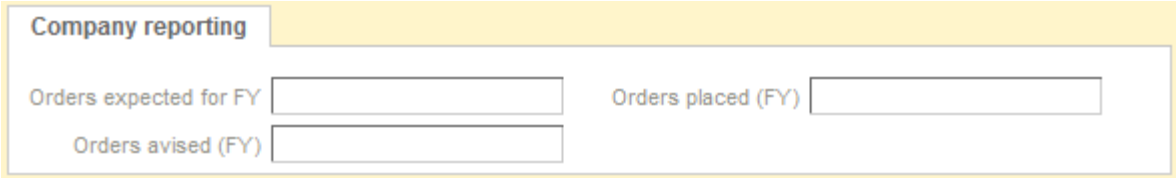
release. This configuration provides backward compatibility so that the layout of existing installations will not change in the update.

Additionally the administrator does have the flexibility to remove watermarks or tooltips from the individual data object group fields in a unit group. The annotations on field level have a higher priority so it always is possible to define different behavior for just a single field.

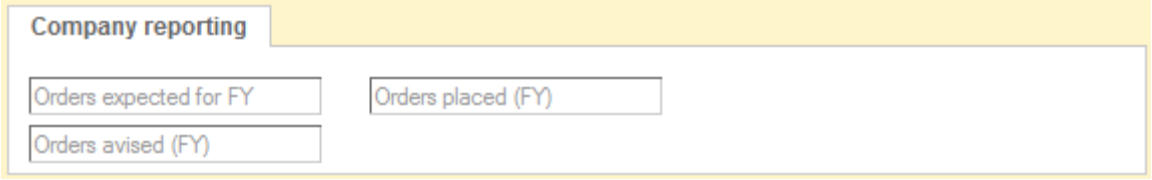
All new data object groups will be rendered with the new default configuration. This means:

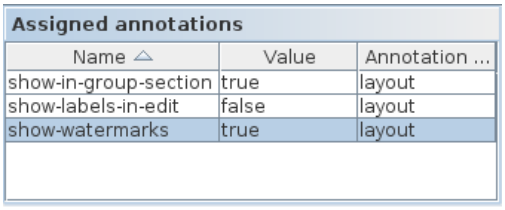
- standard labels (unlike previously added custom fields)
- watermarks disabled
- tooltip enabled

Please be aware that this is a new default behavior! The following screenshots illustrate this. The first screenshot shows the new behavior without additional configuration. The fields of the new group have labels, but show no watermark.



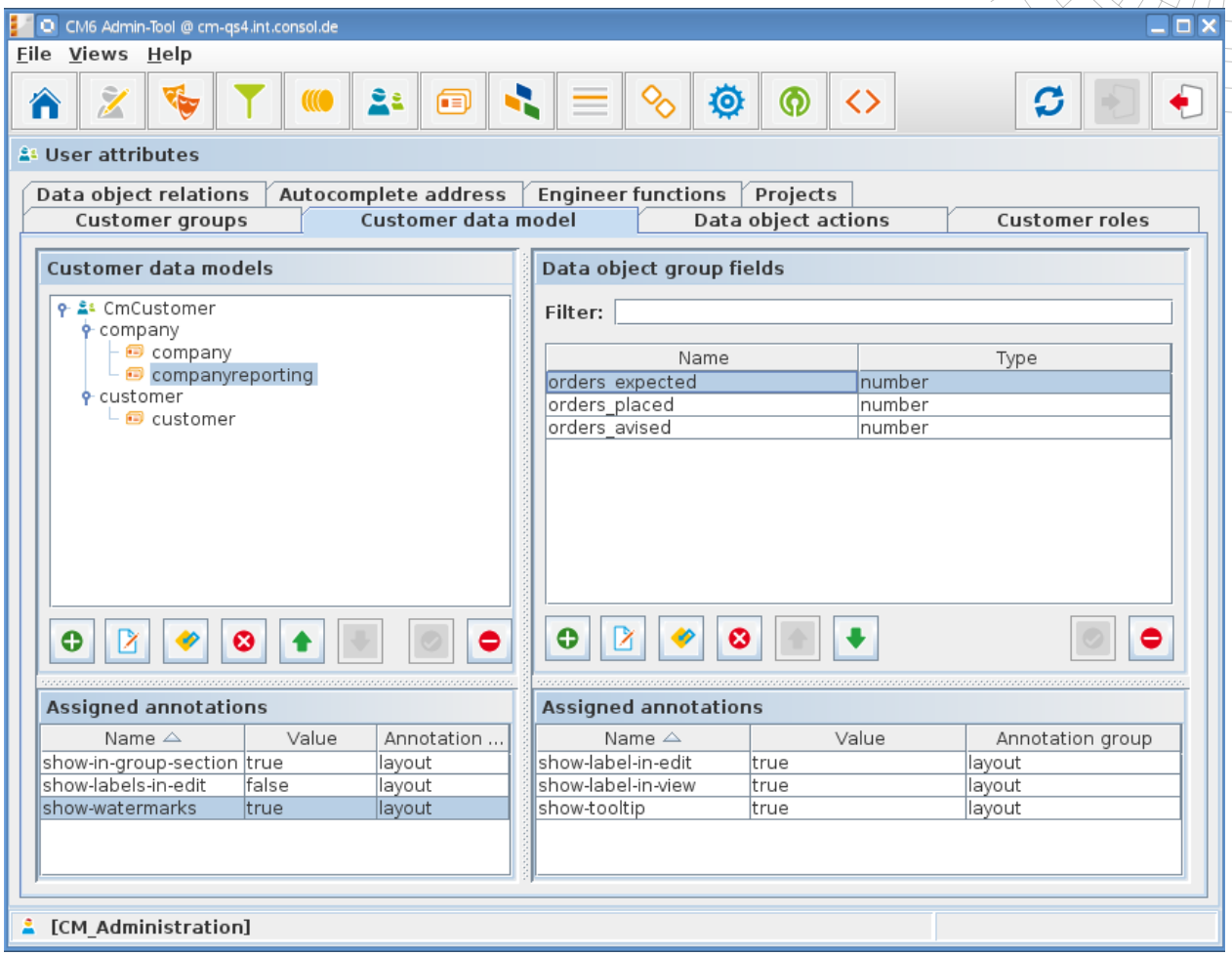
The previous behavior for all the fields in the group can be achieved by the group level annotations shown below.



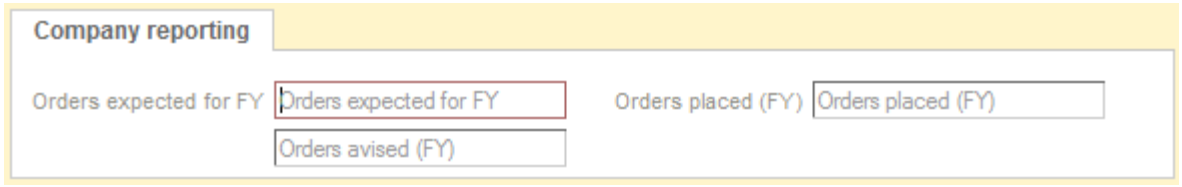


The annotation `show-labels-in-edit` has been set to *false* so that no labels are shown and `show-watermarks` has been set to *true* so that watermarks are displayed instead.

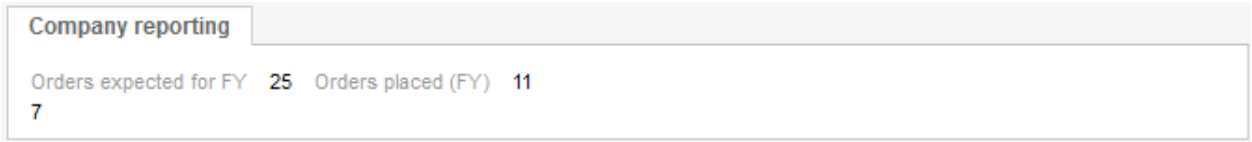
The fields can be controlled individually to override the group level settings. This has been done in the example below with the fields “orders_expected” and “orders_placed” from the data object group “companyreporting”. The field-level annotations `show-label-in-edit`, `show-label-in-view` and `show-tooltip` have been set to *true* overriding the group-level annotations. The field “orders-avised” has not been annotated, so it still uses the values set by the group.



As a result there is no label for the last field when editing/creating a company, defined by the group level annotations whereas the other two fields override this by use of the field level annotation `show-label-in-edit`.



The same holds true for the display-only view of the company. The first two fields are shown with label, while the last one is a number only as defined by the data object group annotations.



Generally the layout is determined by the position annotation, if set. In following case, assumed the configuration is set to show labels for the whole group, the positions are defined as listed. Then the data object group field layout will be displayed as in the table below.

- `Field_1: position 0;0`
- `Field_2: position 0;1`
- `Field_3: position 1;0`
- `Field_4: position 1;1`
- `Field_5: position 2;0`
- `Field_6: position 2;1`

`Label_1`	`Field_1`	`Label_2`	`Field_2`
`Label_3`	`Field_3`	`Label_4`	`Field_4`
`Label_5`	`Field_5`	`Label_6`	`Field_6`

In this table each label is displayed in its own table cell which means that additional columns are added for dealing with the labels. The configuration effectively allows for three columns maximum, only technically there would be rendered six columns for adding the labels.

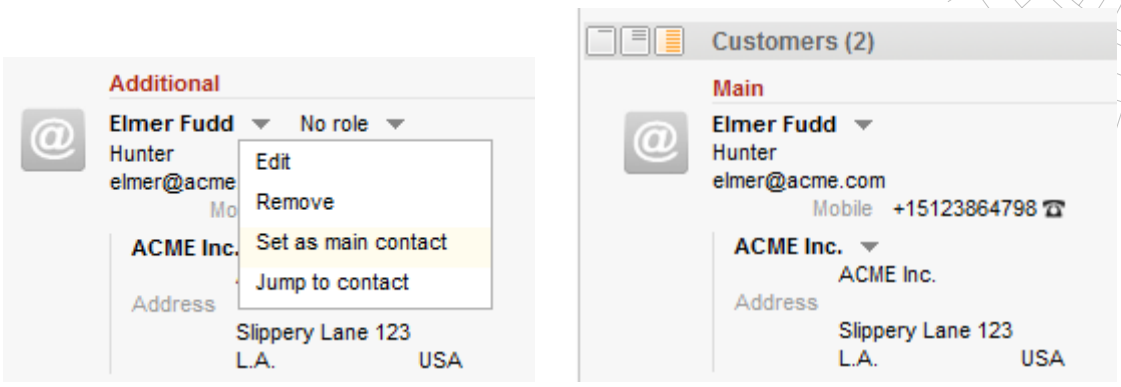
When changing only Field_3 to be displayed without label, setting `show-labels-in-edit=false`, the result will be a layout as shown in the following table:

`Label_1`	`Field_1`	`Label_2`	`Field_2`
	`Field_3`	`Label_4`	`Field_4`
`Label_5`	`Field_5`	`Label_6`	`Field_6`

The Layout is behaving this way due to backwards compatibility. The position annotation applies to the field, not the label. The respective label cell is prepended without this reflecting in the position annotation value for the actual field. This holds true even when the label will not be shown.

1.2.4 Web Client Functionality to Set an Additional Customer as Main Customer (#624227, #625017)

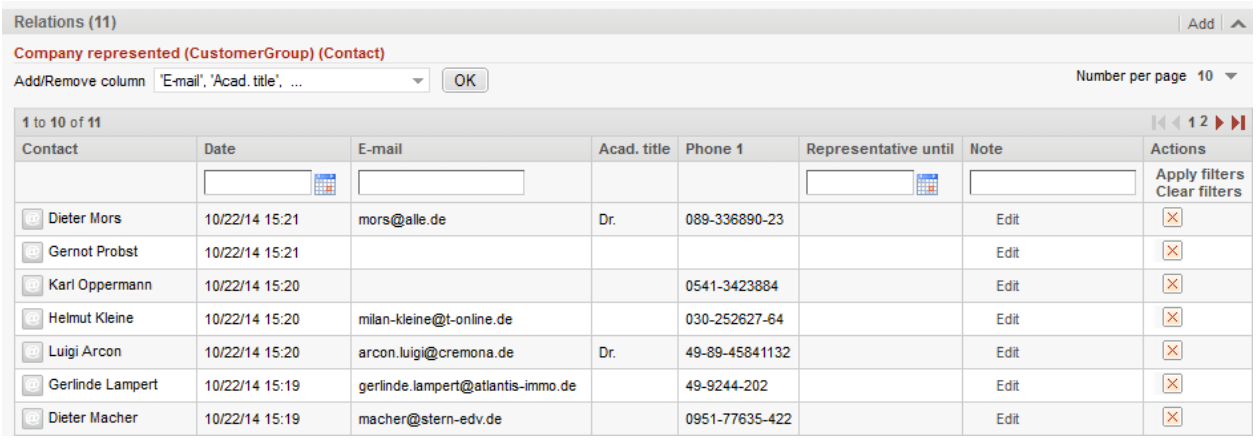
The web client provides a new functionality to define an additional customer as new main customer of a ticket. This is done by clicking the new entry “Set as main customer” in the triangle menu right next to the additional customer’s name. After this the former additional customer will be the ticket’s main customer and the former main customer will be an additional customer. Please compare the screenshots below for an illustration on setting an additional customer to be the main customer, showing an additional customer with the menu entry and the result.



This operation can be undone just by repeating it: The original main customer which shall be restored is now an additional customer. So this function to set it as main customer can be applied again to it. However, be aware that this operation also removes the role set for an additional customer. In case an additional customer was made to the main customer and later becomes an additional customer again by setting another main customer any role assigned initially to the additional customer must be set once again.

1.2.5 Web Client Functionality for Filtering and Sorting Customer Relations by Data Object Group Fields (#625617)

The relations section on the customer pages (company and contact) have been extended with filtering and sorting capabilities, compare the screenshot below.

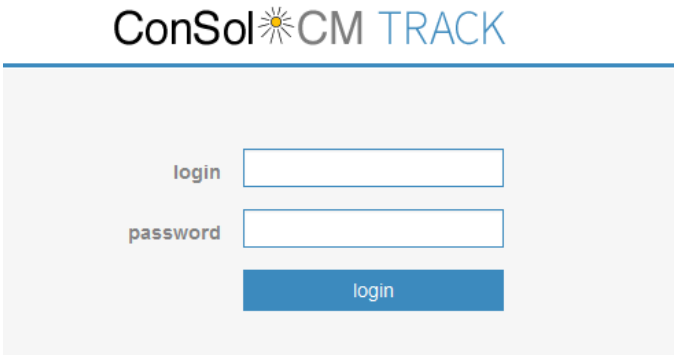


Previously only the date and the note field provided filtering functionality for the element list of a relation. Now every data object group field can be used for filtering. The desired field needs to have the annotation *field indexed* set. The filter options will only be shown if the number of entries exceeds the limit defined in the page customization for the compact subsection display (the limit is set by the value of the attribute *compactViewLimit* in the scope */companyEditPage/unitRelationSection*). This limit defaults to 10, so usually the filters are shown with 11 relation entries or more.

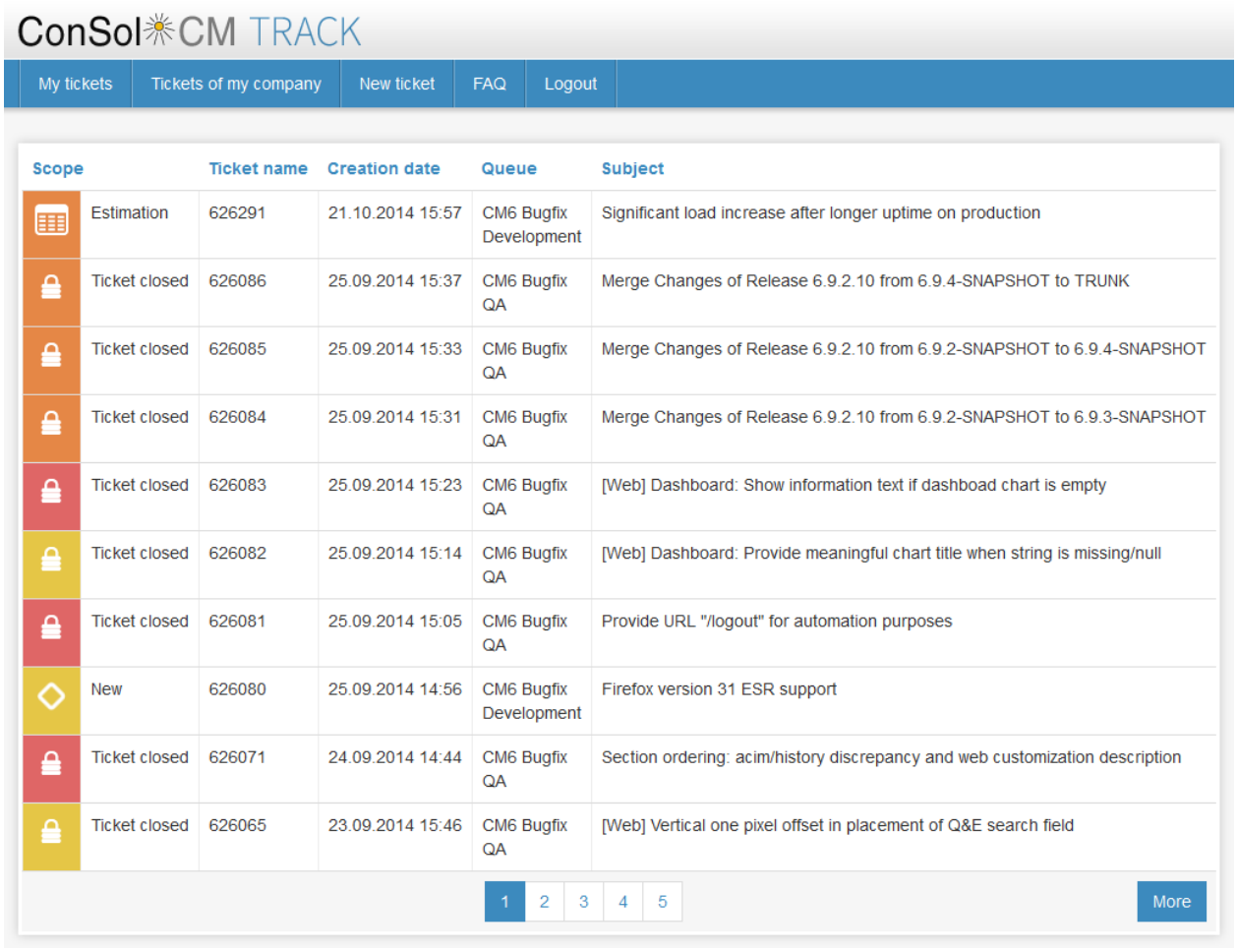
The same annotation *field indexed* controls if the data object field column title can be clicked for sorting the listed entries by its values.

1.2.6 CM/Track: New Visual Theme (#625214)

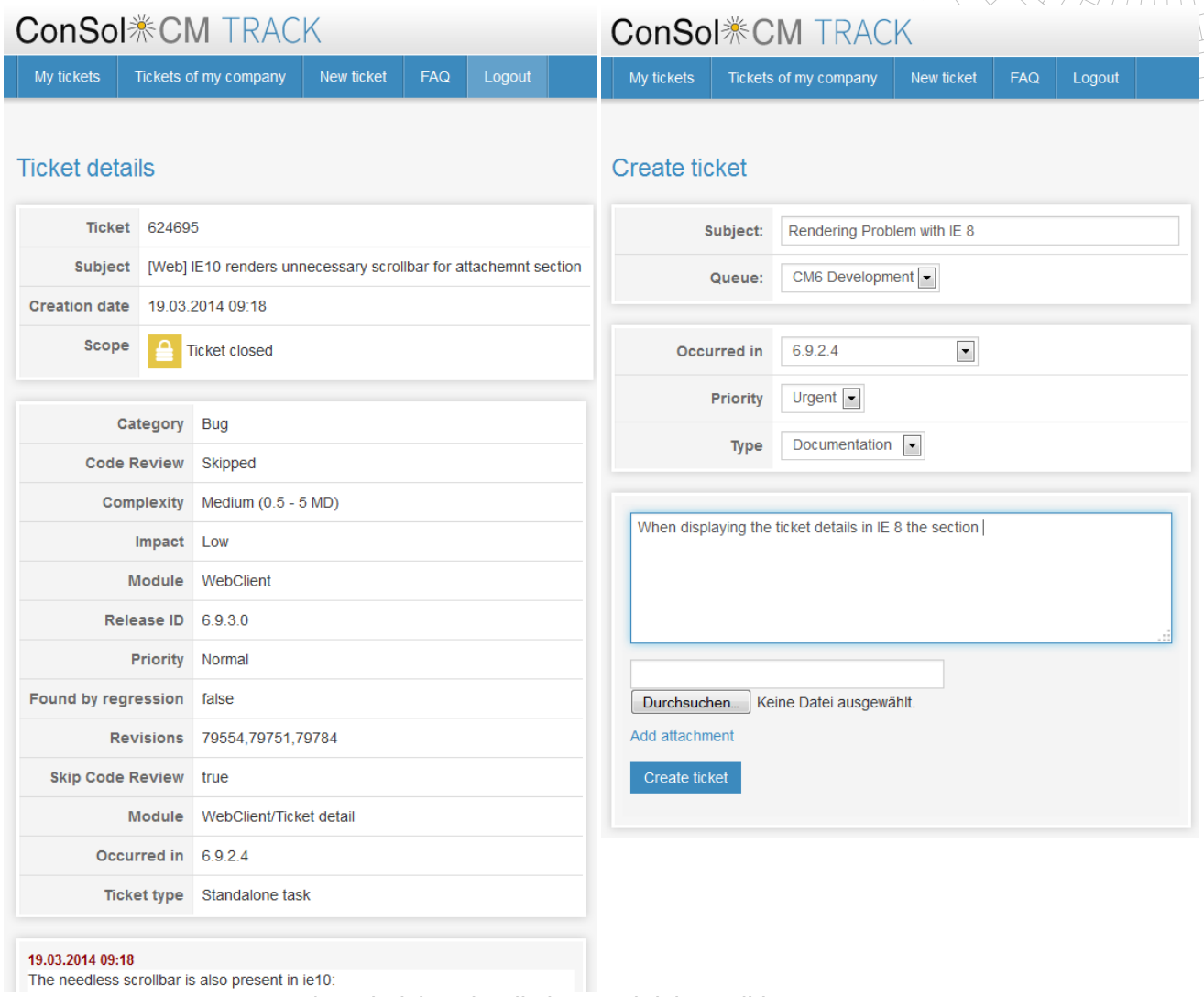
A new standard design theme for CM/Track is being delivered with the current release. It features a more modern look and feel as well as other visual improvements. The following screenshots illustrate this.



CM/Track login page



CM/Track ticket list



CM/Track ticket detail view and ticket editing page

1.2.7 Task Execution Framework (#621709)

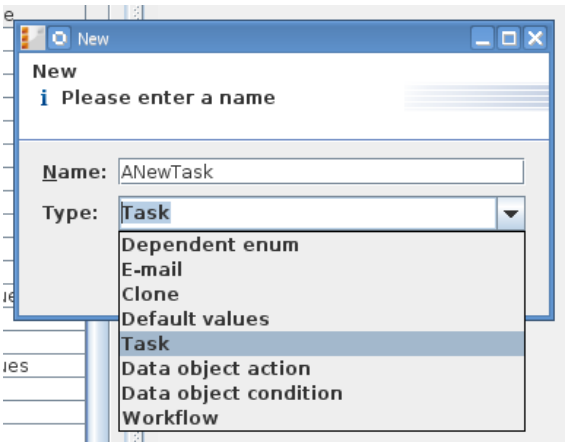
A new feature in this release is a framework for executing dedicated tasks on the system asynchronously. Administrators generally need to have the possibility to execute long running tasks which significantly exceed the regular transaction time boundaries. The newly introduced Task API addresses this requirement. It is available from any extension point of the CM6 core e.g. activity scripts, e-mail scripts or unit actions. Long running tasks are executed asynchronously. It is generally possible to get some basic data of a scheduled task e.g. its start time, progress of task execution or the number of retries done. Tasks in the framework are persistent i.e. they survive a server restart. Tasks can be easily cancelled, even if already started (the level of forcing the cancellation is configurable through method parameters). It is possible to get a report regarding all current active tasks on the server and a dedicated tasks view in the Admin-Tool has been created where the administrator can check the progress/state of the task and cancel it.

Task descriptor: After a task gets scheduled the client code gets a task descriptor handle in return:

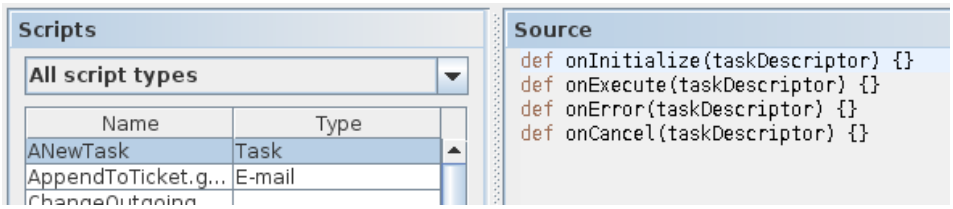
```
TaskDescriptor taskDescriptor = taskExecutionService.execute(someTask)
```

The task descriptor keeps all important information related to task execution and the state of the task. The methods of the task descriptor object can be found in the API documentation.

Tasks executor: The task executor is an engine for tasks execution. It provides a main processing thread (with a watchdog attached) which scans the database for tasks with the status NEW. The second component of the tasks executor is a dedicated thread pool used for task execution. When such a new task is found it is loaded from the database and executed using a new thread with the name set to the task name for better identification, within a new transaction and security context.



Task script creation: A new task script can be created as a Groovy script in the Admin-Tool's *Script and Template Administration* page on the *Scripts* tab. When creating the script in the same way as any other script the type "Task" has to be selected. The basic task function classes are being suggested then in the source code editor field (See below in the section "Task script usage" also). Please see the screenshots for illustration.



Task script usage: The following code examples show the basic procedures of managing tasks in scripts. Creating a task can be done in a script, but it also can be done using the Admin-Tool.

A Groovy task can have a static script associated. Such a static task script, being defined in the Admin-Tool, must meet certain requirements. During the different possible phases of task execution specific methods within the script are executed hence the proper naming of the script methods is crucial. This is the example of an empty script:

```
def onInitialize(taskDescriptor) {}
def onExecute(taskDescriptor) {}
def onError(taskDescriptor) {}
def onCancel(taskDescriptor) {}
```

(a) Creation of a groovy task:

```
GroovyTask groovyTask = new GroovyTask();

// Set a predefined script for the task
groovyTask.setStaticScript(scriptSourceService.getByName("someATScript.groovy"));

// Alternatively a dynamic anonymous script can be set
groovyTask.setDynamicScript(scriptString);

// During creation you also have to set a string called group id (here "taskgroup")
// which allows for easier tasks management
taskDescriptor = taskExecutionService.schedule(groovyTask, "taskgroup");
```

(b) Fetching the task descriptor and cancelling a groovy task:

```
// By having the task descriptor you can obtain all useful task info
TaskDescriptor taskDescriptor = taskExecutionService.getById(descriptorId);
taskExecutionService.cancel(descriptorId)
```

The `descriptorId` used to identify an individual task has to be stored in a variable to access it in the script or in a custom field, if it was created by another script. The `groupId` can be used for labeling a task, but querying for it will return a list of tasks.

(c) Repeating a task:

If you set a new future execution date for a task after its core operations have been executed, the task will be rescheduled (the example adds 15000 seconds, about 4.15 hours):

```
def onInitialize(taskDescriptor) {}

def onExecute(taskDescriptor) {
    // Some task code to execute
    ...
    // Set the new future execution date for the task,
    // you also need to return the special controlling object
    taskDescriptor.setExecutionDate(new Date(new Date().getTime() + 15000));
    return new ExecutionSpecification().setRetryRequested(true);
}

def onError(taskDescriptor) {}
def onCancel(taskDescriptor) {}
```

(d) Repeating a task after an error occurred:

```
def onInitialize(taskDescriptor) {}

def onExecute(taskDescriptor) {
}

def onError(taskDescriptor) {
    return new ExecutionSpecification().setRetryRequested(true); // This will
    // reschedule the task for immediate reexecution, in case a future date
    // is needed when rescheduling, it can be set like above for regular repetition
```

```
}
def onCancel(taskDescriptor) {}
```

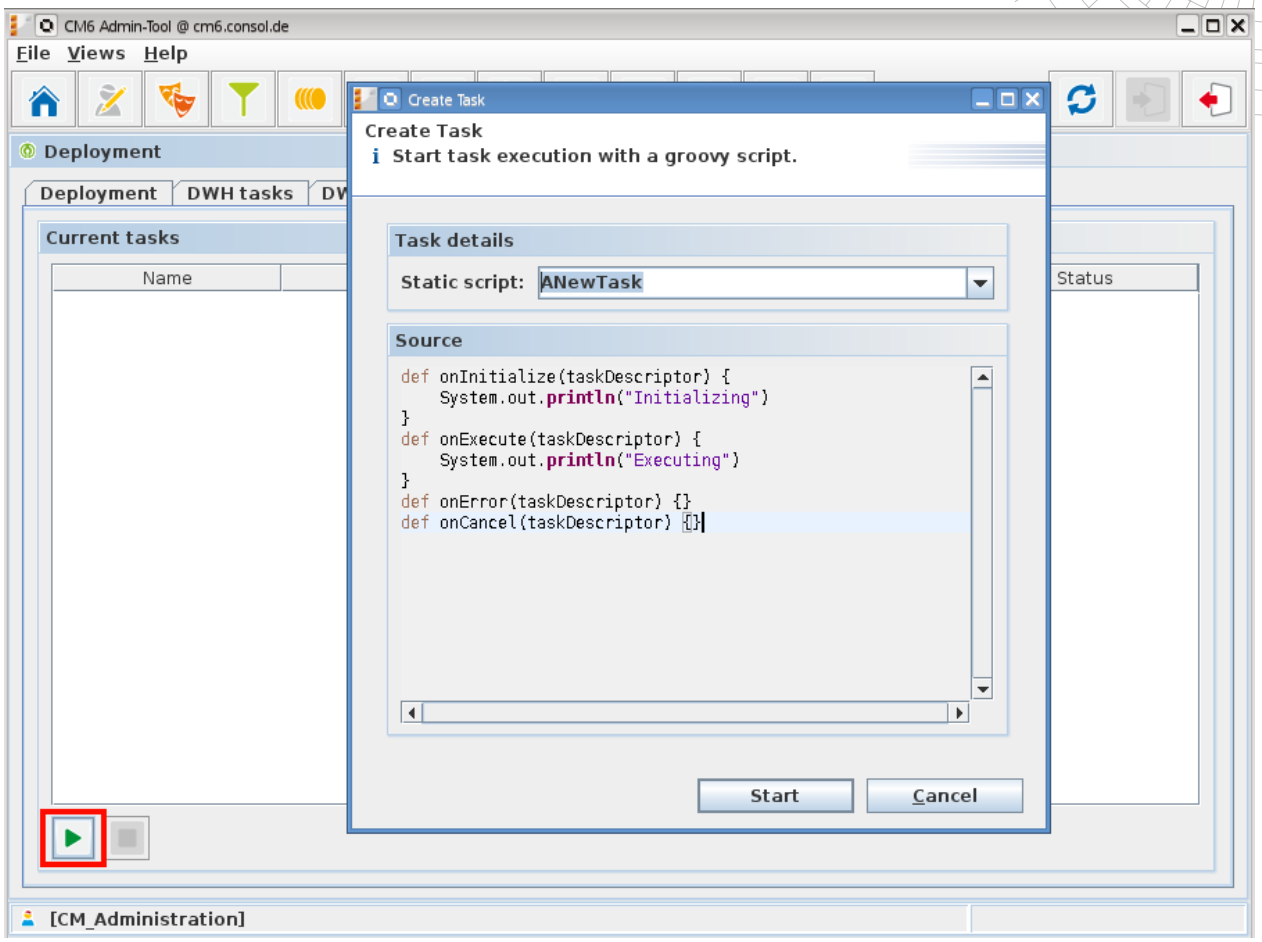
(e) Updating the progress of a task:

```
def onInitialize(taskDescriptor) {}

def onExecute(taskDescriptor) {
taskExecutionService.updateProgress(taskDescriptorId, 55) // any value could
    // be entered here, but AT will narrow it to max 100 anyway
}

def onError(taskDescriptor) {}
def onCancel(taskDescriptor) {}
```

The integer value supplied can be greater than 100, but such a value will be interpreted as 100 (per cent). Effectively it is most reasonable to use values between 0 and 100 for progress.

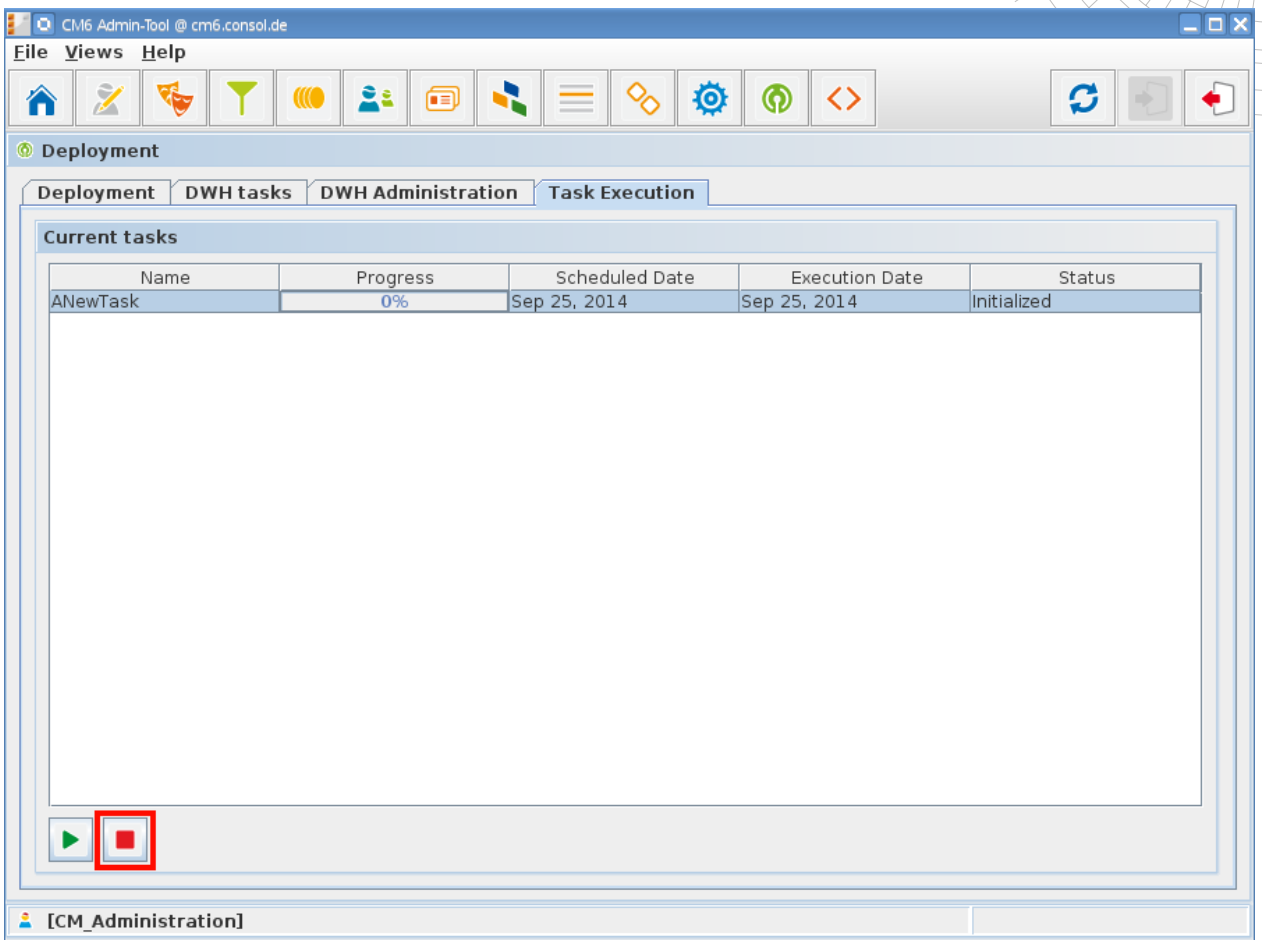
Task execution: In the Admin-Tool page *Deployment* a new tab has been added labeled *Task Execution*. This can be used for starting and stopping task scripts as well as to monitor the progress.

For being able to run scripts it is required to enable the *Start task* button which is hidden by default. This is done by setting the system property *start.groovy.task.enabled* in module *cmas-app-admin-tool* to the value *true*. In case this system property has to be created it must be of boolean type.



Once the *Start task* button is visible a click on it will open the *Create task* dialog to select the script from the available task scripts defined in the *Script and Template Administration* (tab *Scripts*) as described above.

A click on the *Start* button in the dialog will start the task and show it in the list of the *Task Execution* tab. The running task can be selected in this list and it will be cancelled by clicking on the *Stop task* button below the list on the tab.



1.2.8 Annotation Modification during Scene Import (#625243)

A change has been introduced which allows clients to participate in jar transfer update. With this mechanism a customer specific project can manage annotations (add/change/rename) during the scene import after an update. This is mainly relevant while creating new custom CM6 systems.

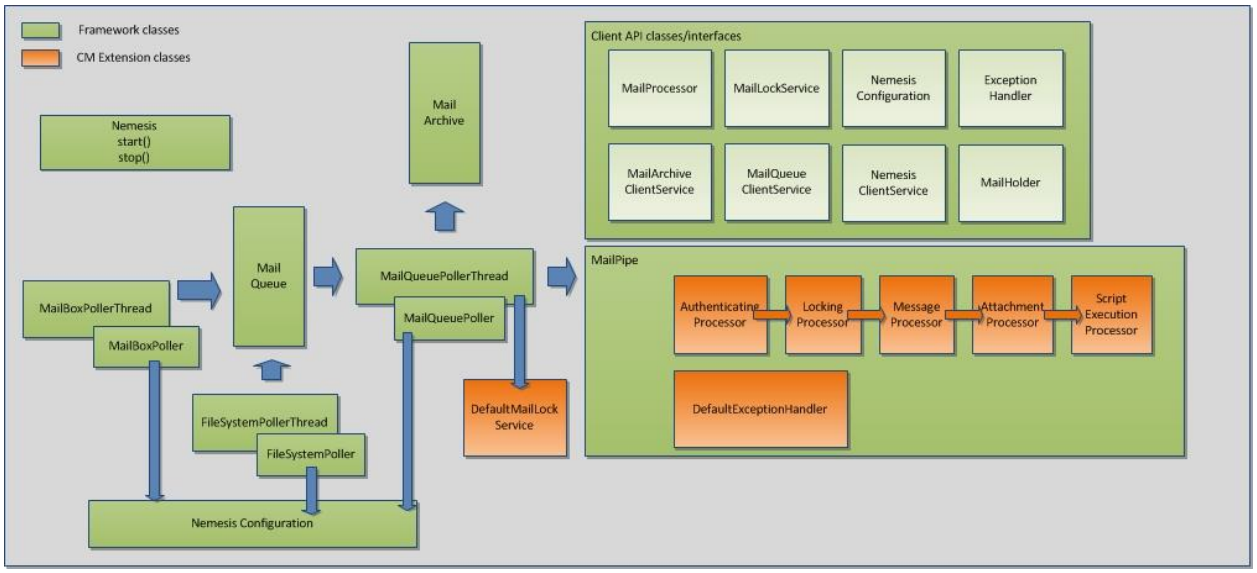
1.3 Changes

1.3.1 Layout Improvements

- **Boolean fields representation unified in web client (#625478):** A boolean field with no value set (yet) will not be shown any more in view mode. This is different compare to the display mode in previous versions which showed it with the value *no*, even though no value was set (i.e value *null*). The field will be shown with value *no* only, if this value has been set explicitly before. In edit mode checkboxes and radio buttons which are unchanged will have no values set on submit. After a change they will have been set to either a value *yes* or *no*. A boolean field that is shown as select list in edit mode can be reset to an unset value by selecting the *Choose one* entry.
- **Buttons for adding comments/sending e-mails change alignment to left (#625193):** The buttons *OK* and *Cancel* for the rich text editor when entering comments and e-mail body text have been moved to the left so that they align with the left edge of the editor field. This improves usability since the buttons show up in the same place independently of the (possibly resized) editor field.
- **New sort order of column selection for result tables (#624808):** The column selection dropdown lists for the search page results had not been sorted properly in previous releases. For every result section on the contact and company pages and the details search page the columns are now sorted. The "checked" columns selected for display show up first in the order of display and all other available columns are listed afterwards in alphabetical order.

1.3.2 NIMH New Incoming Mail Handler (#620992)

NIMH general architecture: NIMH is the new mail processing framework that comes with CM6. It comprises a set of interfaces, core services and threads for mail fetching from different sources and processing those in a dedicated pipe. By default NIMH does not provide any mail processors - it is up to the framework clients to provide implementations. The following diagram uses Nemesis as a framework name, please consider substituting Nemesis with NIMH.



1. **MailBoxPoller**.
   Starts and manages threads with logic polling mails from configured mailboxes. Can handle different type and number of mailboxes i.e. IMAP(S) and POP3(S), examples of configurations can be found below. Configuration changes are reloaded on the fly without the need of a server restart.
2. **FileSystemPoller**
   Starts and manages thread responsible for file system polling. This feature loads mail in form of .eml file from given (configured) location in file system, which is very useful for debug and development.
3. **MailQueue**
   Mail retrieved from any source (real mailbox or filesystem) is put into the database, into a special table, which acts like a queue of mails waiting for processing. On the other end of the queue, special a thread pools e-mail by e-mail and puts it into the MailPipe where the actual processing is done.
4. **MailQueuePoller**
   Special component which manages threads which work as consumer of e-mails put into the MailQueue. The e-mails are loaded one by one and they are processed using MailPipe. The number of processing threads can be increased if needed. If there has been an error during e-mail processing, the thread will try to load and process this mail several times more and if still cannot produce a successful result, the e-mail will be moved to MailArchive.
5. **MailArchive**.
   Another database structure for keeping mails which were processed with errors. The file system directory is no longer used to store such unparsable e-mails.
6. **MailPipe**
   Pipe comprising processors which do the actual mail processing. They all need to implement NIMH framework interface MailProcessor and are executed as a chain where each processor must explicitly invoke next the one in the chain otherwise the execution chain will be broken (sometimes this can be intentional e.g. for e-mail filters).
7. **MailProcessor**
   Main interface for client implementations where code responsible for real business e-mail processing is located.
8. **MailLockServcie**.
   Another extension point of the NIMH framework. Client code can provide custom implementation of MailLockService interface. If provided, a lock must be obtained before an e-mail will be processed in the MailPipe. For example in CM core reality, the default lock service implementation lock uses the ticket name, other words, for a given ticket only one e-mail can be processed at a time
9. **Nimh.**
   Top level service tool for stopping/starting all of the NIMH services
10. **ExceptionHandler**
    Called by the MailPipe when an exception is thrown from the processors chain. usefull for specific handling tasks like sending e-mails from processing etc

NIMH configuration: The idea of configuration of NIMH is based on convention. There is a default set of all configuration properties having default values. They all start with *mailbox.default* prefix. These default values can be overwritten by values specific for a particular mailbox prefix e.g. ***mailbox.myPop3Mailbox*.** All properties starting with the prefix: ***mailbox.default.session*** will be assumed as *java.mail* properties and passed into java mail session**.**

1) MailMailboxPoller
   a) **mailbox.polling.threads.number** - Number of threads for accessing mailboxes.
      Default value: 1
   b) **mailbox.default.task.delete.read.messages** – This defines whether messages should be removed from the mailbox after processing. For IMAP protocol messages are marked as *SEEN* by default. For POP3 protocol, when flag is set to *true* the message is removed, otherwise remains on server and will result in infinite reads.
      Default value: false
   c) **mailbox.default.task.interval.seconds** – Default interval for polling mailboxes.
      Default value: 60 seconds
   d) **mailbox.default.task.transaction.timeout.seconds** – Default transaction timeout for mail fetching transactions. Should be correlated with the number of messages fetched at once.
      Default value: 60 seconds
   e) **mailbox.default.task.timeout.seconds** – After this time (of inactivity) the service thread is considered as damaged and automatically restarted.
      Default value: 120 seconds
   f) **mailbox.default.task.max.messages.per.run** – Number of messages fetched at once from mailbox. Must be correlated with transaction timeout.
      Default value: 20
   g) **mailbox.default.task.max.message.size** – Maximum size of e-mail messages in bytes. Messages with a size greater than the value of this property can be processed from the Admin-Tool.
      Default value: 10485760 (10 MB)
   h) **mailbox.default.task.enabled** – With this property a service thread related to given poller can be disabled.
      Default value: true
   i) **mailbox.default.connection.protocol** - Poller's protocol e.g: IMAP or POP3.
      No default value
   j) **mailbox.default.connection.host** – Host (server name) of a given mailbox from which the poller reads e-mails.
   k) **mailbox.default.connection.port** – Port for a given mailbox from which the poller reads e-mails.
   l) **mailbox.default.connection.username** – Username for a given mailbox from which the poller reads e-mails.
   m) **mailbox.default.connection.password** – Password for given mailbox from which the poller reads e-mails.
   n) **mailbox.default.session.mail.mime.address.strict** – Example javax.mail property - counterpart of the old mule mail.mime.strict, allows to set not so strict mail header parsing.
   o) **mailbox.default.session.mail.debug** – Example javax.mail property - allows for more detailed javax.mail session debugging.
   p) **mailbox.default.session.mail.imap.timeout** – Example javax.mail property.
   q) **mailbox.default.session.mail.imap.connectiontimeout** – Example javax.mail property.
   r) **mailbox.default.session.mail.pop3.timeout** – Example javax.mail property.
   s) **mailbox.default.session.mail.pop3.connectiontimeout** - Example javax.mail property.

2) MailQueuePoller
   a) **queue.polling.threads.number** – Number of threads started for mails' queue polling.
      Default value: 1
   b) **queue.polling.threads.shutdown.timeout.seconds** – Waiting time after the shutdown signal. When the timeout is reached, the thread will be terminated.
      Default value: 60 seconds
   c) **queue.polling.threads.watchdog.interval.seconds** – Watchdog thread interval.
      Default value: 30 seconds
   d) **queue.task.interval.seconds** – Main mails' queue polling thread interval.
      Default value: 15 seconds
   e) **queue.task.transaction.timeout.seconds**. – Transaction timeout for mail processing in the pipe.
      Default value: 60 seconds
   f) **queue.task.timeout.seconds** – After this time (of inactivity) the service thread is considered as damaged and automatically restarted.
      Default value: 600 seconds
   g) **queue.task.max.retries** – Maximum number of e-mail processing retries after an exception. When reached, the e-mail is moved to the e-mail archive. This e-mail can be rescheduled again using NIMH API (or the Admin-Tool).
   h) **queue.task.error.pause.seconds** – Maximum number of seconds, the queue poller waits after infrastructure (e.g. database) error.
      Default value: 180 seconds
3) FileSystemPoller
   a) **filesystem.polling.threads.number** – Number of threads started for mails' queue polling.
      Default value: 1
   b) **filesystem.polling.threads.shutdown.timeout.seconds** – Waiting time after the shutdown signal. When the timeout reached, thread will be terminated.
      Default value: 60 seconds
   c) **filesystem.polling.threads.watchdog.interval.seconds** – Watchdog thread interval.
      Default value: 30 seconds
   d) **filesystem.task.interval.seconds** – Default interval for polling mailboxes.
      Default value: 60 seconds
   e) **filesystem.task.transaction.timeout.seconds** – Default transaction timeout for mail fetching transactions. Should be correlated with the number of messages fetched at once.
      Default value: 60 seconds
   f) **filesystem.task.timeout.seconds** – After this time (of inactivity) the service thread is considered as damaged and automatically restarted.
      Default value: 120 seconds
   g) **filesystem.task.enabled** – With this property the service thread related to the given poller can be disabled.
      Default value: true
   h) **filesystem.task.polling.folder** – Polling folder location which will be scanned for emails in the format of eml files.
      Default value: \home\cmas

4) Nimh-extension:
   a) **[cmas-nimh-extension | mail.ticketname.pattern]** – This is an equivalent to the old **[cmas-esb-mail | mail.callname.pattern]**
   b) **[cmas-nimh-extension | mail.attachments.validation.info.sender]** – This is an equivalent to the old **[cmas-esb-mail | mail.attachments.validation.info.sender]**
   c) **[cmas-nimh-extension | mail.attachments.validation.info.subject]** – This is an equivalent to the old **[cmas-esb-mail | mail.attachments.validation.info.subject]**
   d) **[cmas-nimh-extension | mail.db.archive]** – This is an equivalent to the old **[cmas-esb-mail | mail.db.archive]**
   e) **[cmas-nimh-extension | mail.error.from.address]** – This is an equivalent to the old **[cmas-esb-mail | mail.mule.service]**
   f) **[cmas-nimh-extension | mail.error.to.address]** – This is an equivalent to the old **[cmas-esb-mail | mail.process.error]**
   g) **[cmas-nimh-extension | mail.on.error]** – This is a new property. When set to *true* an error mail is sent to the above configured address in case email message could not be processed.

Mapping of former MULE and new NIMH properties:

MULE property	NIMH property
(cmas-esb-mail)mail.delete.read	(cmas-nimh)mailbox.default.task.delete.read.messages
(cmas-esb-mail)mail.polling.interval	(cmas-nimh)mailbox.default.task.interval.seconds
(cmas-esb-mail)mail.process.retry.attempts	(cmas-nimh)queue.task.max.retries
(cmas-esb-mail)mail.mime.strict	(cmas-nimh)mailbox.default.session.mail.mime.address.strict
(cmas-esb-mail)mail.encryption	(cmas-core-server)mail.encryption (moved to core server properties)
(cmas-esb-mail)mail.callname.pattern	(cmas-nimh-extension)mail.ticketname.pattern
(cmas-esb-mail) mail.attachments.validation.info.sender	(cmas-nimh-extension)mail.attachments.validation.info.sender
(cmas-esb-mail) mail.attachments.validation.info.subject	(cmas-nimh-extension)mail.attachments.validation.info.subject
(cmas-esb-mail)mail.db.archive	(cmas-nimh-extension)mail.db.archive (seems not legit now)
(cmas-esb-mail)mail.mule.service	(cmas-nimh-extension)mail.error.from.address
(cmas-esb-mail)mail.process.error	(cmas-nimh-extension)mail.error.to.address

Please note that the property *mail.encryption* has been moved from the module *cmas-esb-mail* to the module *cmas-core-server* generally! This is not only valid for NIMH but also applies to MULE from this release on (#625494).

NIMH extension points

1) MailProcessor
2) ExceptionHandler
3) MailLockService

NIMH and MULE coexistence: For some limited time MULE mail handling will still be supported for CM6. MULE is still the default choice for new installations. However, starting with this release 6.9.4.0 the Admin-Tool mail configuration (*Configuration* page, tab *E-Mail*) will save the configuration in both formats i.e. as NIMH properties also. This allows to switch to NIMH (and back to MULE, if needed) at any time just by setting the new dedicated configuration

property **cmas-core-server.nimh.enabled**=true. The switch is done automatically during runtime, there is no need for a server restart.

NIMH in a clustered environment: The capabilities of NIMH in a clustered environment are do not differ from MULE for this release. This means you can configure and run it only on one node in a cluster. Cluster capabilities of NIMH will be extended in future releases. All other nodes, not running NIMH, also must have the ESB service disabled for proper operation. To use NIMH in a CM6 cluster one needs to enable it on the desired node. NIMH is enabled on the node using a clustering configuration property instead of the standard property mentioned in the previous paragraph:
cmas-core-server.nimh.enabled.CLUSTER_NODE_ID=true
for example: **cmas-core-server.nimh.enabled.**1=true

Update of previous e-mail scripts: While properties are created an mapped automatically during an update, some variables usually used within mail scripts have changed. If there were modifications to standard mule groovy scripts, they have to be applied to NIMH scripts as well:

Mule variables	NIMH variables
mailContextService	mailSupportService
msg	mailHolder
mailLog	mailLog
spring cm services	spring cm services
n/a	pipeContext (used as parameter in mailSupportService invocations)

The scripts also need to have specific names – The prefix "Nimh" has to be added:
1) *NimhAppendToTicket.groovy*
2) *NimhCreateTicket.groovy*
3) *NimhIncomingMailRouting.groovy*
4) *NimhMailToClosedTicket.groovy*

1.3.3 JMS Redelivery Configuration under JBoss 5 and Weblogic (#625014)

JMS messaging setting for the redelivery interval has been changed to 60 seconds. For exisiting Weblogic installations we suggest to configure weblogic jms message redelivery for the cm6-index and the cm6-mail queues.

Suggested values for Weblogic are:

```
<delivery-params-overrides>
    <redelivery-delay>60000</redelivery-delay> <!-- 60 seconds -->
</delivery-params-overrides>
<delivery-failure-params>
    <redelivery-limit>-1</redelivery-limit> <!-- infinite times -->
    <expiration-policy>Discard</expiration-policy>
</delivery-failure-params>
```

1.3.4 Deactivation of Ticket Deletion in the Admin-Tool by Property (#624439)

A new property has been added which prevents (accidental) deletion of tickets in the Admin-Tool. When this configuration is made, the “Delete” button below the ticket administration ticket list is not shown and the corresponding context menu entry cannot be selected. Deleting a ticket by pressing the “Delete” button on the keyboard is not possible either.
This can be configured by setting the property **`delete.ticket.enabled`**=`false`. The property is part of the module `cmas-app-admin-tool`.

1.3.5 Disallowed Import of config.jar from Newer Version to Older System (#624760)

The import of a configuration JAR file which was created with a newer version of CM6 Admin-Tool than installed was not disabled earlier. Now such an import will be prevented yielding a warning which informs that the import format version is newer than supported by this installation.

1.3.6 No Restart Required after Changing server.session.timeout.SOURCE_ENUM_VALUE (#624606)

Changing the property `server.session.timeout.SOURCE_ENUM_VALUE` did not invalidate the session timeout cache. As a result a server restart previously was required for this property value change to be effective. This has been changed so that the new value is used immediately without restarting the server.

1.3.7 Support 'log' Queue in Direct Database Communication Channel for Data Warehouse (#624414)

The data warehouse communication channel can be set to mode “DIRECT” which uses database communication. In this case now the log information is being read from and written to this communication channel (technical value `INT_LOG_QUEUE`), too, instead of the JMS queue formerly. The JMS queue is not available for JMS data warehouse communication any more.

1.3.8 Deactivating units: Missing API Method to Deactivate and Replace a Unit (#624714)

Previously there was a method missing in the unitServiceAPI which was able to deactivate a unit after replacing it with another unit in all tickets. Due to concurrency behavior it was not easily possible to use separate methods for these two steps in conjunction in a script. A new method `UnitService.replaceAndDeactivateUnit(unit source, unit target)` was introduced to provide this functionality:

```
void replaceAndDeactivateUnit(Unit pUnit, Unit pReplacement)

Replaces given unit in all of its open tickets, then deactivates it with no
cascade to associated units
Parameters:
pUnit - Unit to replace and deactivate
pReplacement - Unit's replacement
```

1.3.9 Sorting Activities in REST API (#625008)

Previously ticket activities had been returned unordered to the client by the REST API. This has been changed and the activities are now returned in same order as in the web client.

1.3.10 Retrieving the Default Text Entry Class by type via REST API (#625147)

A new REST API call has been implemented in order to be able to get the default entry class for the history and attachments entries. In order to get the default entry class by the type the following requests have to be made:

```
GET /entryclasses/default/{type}     (Get default entry class by type.)
```

Path param

`type`	Default entry class type (supported: ATTACHMENT, TEXT)

Query param

`v`	Version of element [optional]

Header param

`X-Template`	Template to use to prepare output data
`rest_response_entryclass`	Default

Format

`JSON`
`XML`

Cache

`FOREVER`	A year from now if version parameter is set
`NO_CACHE`	Do not allow any caching

Returns

`200`	Entry class was loaded successfully
`303`	See other if given version is stale
`404`	There is no default entry class for specified type

1.3.11 Improvement of the Localization Property for RegEx Validation (#625109)

The validation message when checking a field value against a regular expression (matches annotation) has been improved to be better understandable for a user. It will not show the regular expression anymore.

1.3.12 New URL "/logout" for Automation Purposes (#626081)

A special URL in the web client is being introduced with this release:

```
http://cm6-server/cm-client/logout/
```

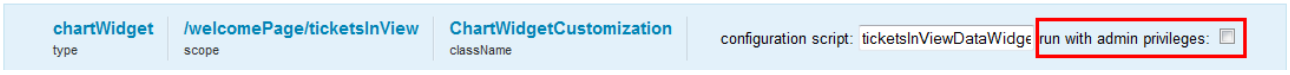
This URL eases automation use cases like monitoring that require login and logout to the web client. It provides a robust way to logout of an automated web client session again not to unnecessarily consume licenses for monitoring.

1.3.13 Page Customization Differentiating Type and Scope Related Properties (global and local, #624910)

A new mechanism has been integrated which can give the user information, if a page customization should be in global or local scope. This is first being used for the new dashboard. More elements will use this in future releases.

1.3.14 Page Customization Scripts Executed in the Context of the Current Engineer (#625669)

Scripts assigned to page customizations previously were executed with admin privileges. In some cases this caused some methods (e.g. ticket search by view) to return undesired or wrong results. Scripts assigned to component customization are executed for the current engineer logged in and must use his permission model. This has been implemented and `ScriptProviderService` and `ScriptExecutionService` no longer require global admin privileges. Starting with version 6.9.4.0 every authenticated user can execute scripts (but only the ones prepared by the admin as `ScriptSource` entities). For backwards compatibility all scripts in use before updating to this release will execute with global admin privileges. The page customization user interface now has a new checkbox to decide whether customization script should be executed with the additional admin privileges. Some scripts may require additional rights, others may rather depend on exact engineer authorization context like the scripts used for dashboard widgets.



1.3.15 Wicket Web Framework Update to Version 6.16.0 (#624611, #625382)

The wicket frame work for web page creation in the web client has been updated to the new version 6.16.0 in order to improve performance and maintainablilty.

1.3.16 CM/Phone: Changed Path for user.config File for Better Use on Terminal Server Environment (#625311)

The file system location of the CM/Phone configuration has been changed so that it can be used on a terminal server environment without additional intervention. Additionally this change offers better access for administration.

1.3.17 Unnecessary Exceptions in server.log from PhoneServiceImpl Removed (#625529)

The log files contained numerous unnecessary exception messages when using CM/Phone without subscriber pattern configuration. A missing check for empty subscriber patterns has been added and the exception messages will not be appearing in the log files anymore.

1.3.18 Support for CSV in the x-forwarded-host HTTP Header (#625376)

The REST API has been extended to accommodate the requirement of the "X-Forwarded-Host" header supporting multiple hosts in complex environments. This header now supports multiple comma-separated values and the proxy IP address can be found in the list received.

1.3.19 Limit of Maximum 1000 Queues Removed (#625870, #625733)

The system had a limit when dealing with more than 1000 queues. Some operations were not able to handle such a situation. The functionality dealing with queues has been changed so that this limit does not exist anymore and many more queues can be handled.

1.4 Bugs fixed

Number	Description
620060	CM6 could not start with a defective e-mail server configuration or in case the e-mail server(s) configured were not available. This limitation has been removed in the context of the introduction of the NIMH New Incoming Mail Handler.
621358	A missing index has been added to the database table recording changes to the database schema.
622394	In the Admin-Tool static criteria for an engineer which were available through a view were not removed when the view was unassigned from the engineer. These criteria were visible, grayed out and thus not editable. They were removed only after a data refresh. This undesired behavior has been corrected and these static criteria are not visible anymore after unassigning the view.
622487	After an update custom fields with date and time entry had a layout issue in some cases placing the minute field slightly below the other fields. This issue has been fixed and all date and time fields align horizontally again.
622700	Incoming emails with improper header data for the To-address like mixed quotes caused tickets to be created with unwanted content. Instead of properly treating the mail as faulty a ticket was created containing a stack trace of the mail handler error and the e-mail message as an attachment. This problem has been fixed so that now such a message is parsed successfully and a regular ticket is created as desired.
623106	Occasionally e-mails with special features could cause some notification by MULE and an ArrayIndexOutOfBoundsException that had been logged. A correction for this issue has been provided in case a switch to NIMH cannot be made to avoid this problem.
623361	In some cases the text display in the rich text editor for comments and e-mails did not keep the format setting chosen manually, but reverted to default settings. This could happen for newly entered text as well as for text already entered before. This highly undesired behavior has been corrected so that the text format settings are now kept in a consistent manner.
623917	The template administration in the web client did not show templates referring to custom fields which were not available anymore. This behavior also prevented correcting the templates and has been fixed. Now templates with invalid placeholders will be colored red and can be edited for correction.
624397, 625922	The drag handle/”Infobox” moving along with the mouse cursor during a drag-and-drop operation was placed wrongly in a specific previous release. It appeared far below the mouse cursor. This undesired positioning has been fixed and the box is displayed right at the mouse cursor again.
624633	A relation between a company-only customer group and another customer-only customer group was labeled wrongly on the customer page. The technical name was shown as label instead of the localized name. This faulty display has been corrected and the localized name is properly used as label now.
624727	In the Admin-Tool the drop down filter list for queues in the view administration was unsorted. This faulty display has been corrected and the entries of this filter list are now sorted alphabetically and case-sensitive.
624732	Several lists in the Admin-Tool did show the selected entry with black font color. This was less readable and inconsistent with white font color used in most lists for the selection. This inconsistency has been removed and all list selections should use white font color now.
624851	The time was shown twice in the web client when the format annotation was used for custom datetime fields and the vale had been set by the REST API. It could not be edited as well then. This has been corrected, doubled time display will not be present any more, the date picker does not show hours or minutes, and setting the value by REST API is described clearly in the REST API documentation.

624884	A customer could end up being saved unwanted in the workspace after editing and saving it. This has been corrected and a customer will not appear in the workspace under these circumstances.
624937	Adding the annotation “dialable” caused an exception when trying to display previously created unit entries with specific value types for the field. This error has been fixed and all customers should be displayed properly after adding the annotation.
625057	In a specific combination of custom fields the phone number provided to resolve to a contact was wrong due to an index problem. This mistake has been corrected and the number should be resolved correctly now.
625065	A phone number field annotated as “dialable” was not indexed transitive, if it was inside of a struct construct. This problem has been generally resolved and the “dialable” phone number string field will be indexed transitive now.
625126	Different functions were used for accepting a ticket depending on the button clicked to accept it. This could cause an exception when used with workflow scripts. This problem was removed and the function used to accept a ticket is uniform now for all ways to perform this action.
625153	An error message was shown for the suggestions when there was a very high number of matching entries for the data provided. This also produced an exception the log files. This has been corrected and a message informing about too many matches is displayed together with a hint to refine the criteria.
625186	During ticket transfer of a scenario import specific attachment data caused an exception in the log files indicating problems indexing these data. The import succeeded anyway. This issue has been remedied and the log exception does not show up anymore.
625205	An exception could occur when quoting an e-mail or a comment containing some special characters. This problem has been solved and quoting works now with all mail and comment content.
625228	After a data warehouse update had not been processed a database error could occur leading to “Value Not Found” and “Incorrect Update Data” errors. This error has been corrected and such a case now only yields a log warning.
625302	Adding new groups and values to an existing MLA could lead to an error while using data warehouse live mode data transfer due to wrongly duplicated IDs. This problem has been resolved, the duplicate ID and errors do not appear any more.
625427	The text classes other than default set for incoming and outgoing emails attached to a ticket were not exported and imported. After an export and re-import of configuration and data these non-standard text-class settings were missing. This deficit has been removed and those non-default text-classes now are present after exporting and re-importing.
625560	It was formerly possible to enable compatibility mode in Internet Explorer for use with CM/Track even though this broke the layout and prevented login. This was changed so that the compatibility mode cannot be enabled for CM/Track anymore.
625769	The web client did not just show an error message informing the engineer when the application server lost the connection to the database. The technical error page of the application server was shown in this case. This has been changed so that in the web client an informational error message is shown which is more helpful for the engineer.
625779	When trying to create a new ticket with an attachment, but providing no entry for a mandatory field, the attachment information was lost after clicking the create ticket button. The reloaded page did not show it anymore. This error has been fixed and the attachment information is preserved in case of this page reload.
625783	In the attachments section of the contact page the file type dropdown was rendered to small and was displayed improperly when opening the list under Internet Explorer version 8. This problem has been corrected and even Internet Explorer 8 displays the dropdown list correctly now.
625952	An engineer was presented an “Edit” link for additional customer data even though no adequate privileges for this customer group were assigned to the engineer. This was corrected and the “Edit” link displays only when sufficient privileges are present.

626003	In the web client ticket transfer from a customer to another customer the watermark hints for the replacement customer search were misleading and partially not in line with the customer data model selected. This misleading labeling has been changed so that useful hints are displayed for all types of customer data models.
626358	Enabling request time logging using the log file "operationtimes.log" on a JBoss 7 environment did not work properly. This issue has been fixed and request time logging can be enabled on JBoss 7 extending the configuration in the proper way for this platform.

2 Version 6.9.4.1 (10.02.2015)

Version 6.9.4.1 includes 6.9.3 versions up to 6.9.3.9, 6.9.2 versions up to 6.9.2.11, 6.8 versions up to 6.8.5.8 and 6.7 versions up to 6.7.13

2.1 Update and installation instructions

2.1.1 Optional minor data warehouse schema update

A minor schema change for the data warehouse was introduced in connection with fixing the issue #626706. In case you want the schema to be fully in line with a fresh installation of a version 6.9.4.1 data warehouse you should execute the update script on the data warehouse database. Applying this optional script, however, is not essential for proper operation of the system.

2.1.2 Optional dashboard script optimization

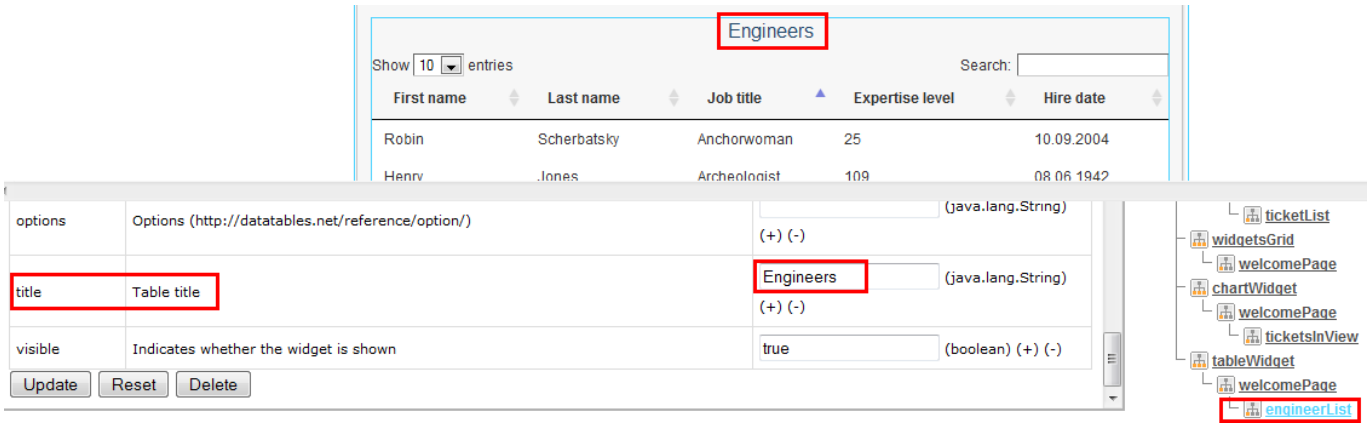
In case you use a CM 6.9.4.0 installation with the default dashboard script, please consider changing this script to the new default (issues #626913, #626909) in order to reflect the performance optimization introduced with this change. The same applies if you use a dashboard script based on the default script. See section 2.3.1 for further details.

No further instructions available.

2.2 New Features

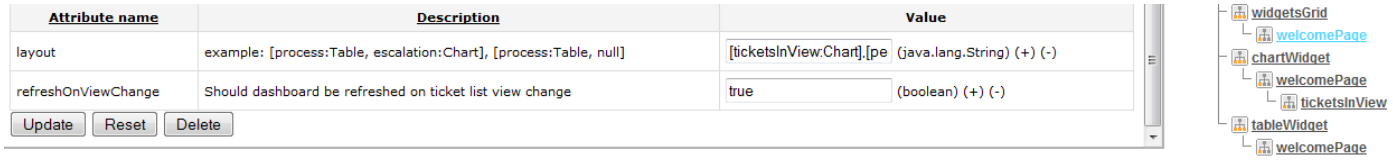
2.2.1 Page customization for defining a title for dashboard table widgets (#626594)

It was not possible previously to define a title/headline for table widgets in the dashboard. This element has been added and for this purpose the page customization attribute *title* has been added for table widgets. It will be available for each table defined separately and it can show localized texts with the same mechanism as other widget options. The example in the screenshot below only uses a discrete non-localized value for illustration, however.



2.2.2 Page customization for dashboard reloading behavior (#626612)

A new page customization has been added to control the reloading of dashboard widget data. Previously the data were reloaded when the view was changed in the ticket list. This reloading is potentially heavy on system performance. Now the reload on view change can be disabled by the new page customization, so that the widget data are only reloaded on a refresh of the whole overview page.
The attribute is called *refreshOnViewChange* and it can be found in the scope *widgetsGrid/welcomePage*.



The attribute default value is *true*. Setting it to *false* will disable reloading the data and refreshing the widgets' display when changing the view in the ticket list of the overview page. This applies to all widgets.

2.2.3 Integration of Highcharts extension library into dashboard (#626590)

The dashboard previously used the Highcharts library for the chart widget without any extension. With this release the addition "highcharts-more.js" providing more chart types has been integrated so that these are available now.

2.2.4 REST API: Creating a contact for an existing company extended (#624876)

Creation of a contact for a company already existing using the REST API has been improved by extending the functionality. Previously it worked for using a reference field to identify the company of the contact to be created. Now the `companyId` can be used as well for company identification. The new parameter and its corresponding example are marked in boldface below.

```
Create unit for given customer group

Form params
customerGroup - customer group of the unit
{group}.{field} - field value
group - name of the custom field group
field - name of the custom field
model - name of unit definition
engineer - name of portal user [optional]
companyId - id of company (unit id) [optional]

Returns
201 - unit has been created
400 - element data are not valid, see validation
404 - customer group doesn't exist
404 - unit definition (model) doesn't exist
404 - portal user doesn't exist

Response header
Location - url to newly created entity

Example
// create a company
curl -u Huber:consol -d
"customerGroup=CustomerGroup&company.name=TestCompany&model=company"
http://localhost:8888/restapi/units
// create a customer for a company using reference field
curl -u Huber:consol -d
"customerGroup=CustomerGroup&customer.name=Test&model=customer&customer.
companyRef=12" http://localhost:8888/restapi/units
// create a customer for a company id
curl -u Huber:consol -d
"customerGroup=CustomerGroup&customer.name=Test&model=customer&companyId
=12" http://localhost:8888/restapi/units

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<location>
  <uri>http://localhost:8888/restapi/units/76</uri>
</location>
```

2.2.5 Supported Firefox now Version 31 ESR (#626080)

Starting with CM release 6.9.4.1 version 31 of the Firefox browser Extended Support Release (ESR) is supported by CM6. The support for the previous ESR version 24 of Firefox is no longer actively maintained.

2.3 Changes

2.3.1 Default dashboard script changed (#626913, #626909)

The default dashboard script for new installations was slightly changed for performance improvements.

The performance-heavy command `ticketService.getIdsByView(viewCriteria).size()` was replaced by the command `ticketService.getCountForView(viewCriteria)` which yields the same result with less load.

The corresponding scenario has been updated. The new script is shown below with the changed sections highlighted in boldface. The script code in the dashboard section 1.2.1 above has been updated as well. In case you are updating from version 6.9.4.0 and using the previous default script, it is strongly recommended to replace it with the new code provided here. Should you use a custom script with the replaced command, please consider also replacing it in your script.

```
import com.consol.cmas.common.model.ticket.*;
import com.consol.cmas.common.model.ticket.view.*;
import java.util.*;
import java.util.Map.Entry;

if (viewId == -1) {
  return [visible: 'false']
}
def engineerLocale = engineerService.getCurrentLocale()
def view = viewService.getById(viewId)
def viewName = localizationService.getLocalizedProperty(View.class,
        "name", viewId, engineerLocale)

ViewCriteria allCriteria = new ViewCriteria(view,
        ViewAssignmentParameter.allTickets(),
        ViewGroupParameter.allTickets(),
        new ViewOrderParameter())
def allTickets = ticketService.getCountForView(allCriteria)

ViewCriteria ownCriteria = new ViewCriteria(view,
        ViewAssignmentParameter.allTickets(engineerService.getCurrent()),
        ViewGroupParameter.onlyOwnTickets(),
        new ViewOrderParameter())
def ownTickets = ticketService.getCountForView(ownCriteria)

ViewCriteria unassignedCriteria = new ViewCriteria(view,
        ViewAssignmentParameter.allUnassignedTickets(),
        ViewGroupParameter.onlyUnassignedTickets(),
        new ViewOrderParameter())
def unassignedTickets = ticketService.getCountForView(unassignedCriteria)

def data = []
data.add("{name: _('all'), data:[${allTickets}]}" as String)
data.add("{name: _('own'), data:[${ownTickets}]}" as String)
data.add("{name: _('unassigned'), data:[${unassignedTickets}]}" as String)

return [series: "[${data.join(',')}]" as String, visible: 'true',
        chart:  "{type: 'column'}", title: "{text: '${viewName}'}" as String,
        tooltip:"{headerFormat:''}" ,
        localization:"de: {all:'Alle',own:'Eigene',unassigned:'Unzugewiesene'},"+
                     "en: {all:'All', own:'Own',   unassigned: 'Unassigned'}"];
```

2.3.2 Dashboard Highchart library updated to version 4.0.4 (#626776)

The Highchart library used for the dashboard chart widgets has been updated to version 4.0.4.

2.3.3 Ticket transfer associated with privilege (#622041)

The function to transfer all tickets associated with a customer to another customer did not require any specific privilege previously. Thus, it could be executed by every user who was able to access the customer. This has been changed and now this function is available only to these users who have the privilege to delete the customer as well. The most common reason to transfer tickets is right before deleting a customer, so this is a natural combination.

2.3.4 Used License count change (#626008)

The modified license count change introduced in CM6 version 6.9.3.6 has been added in this release. It was not present in CM6 version 6.9.4.0 yet. The blocked license count for a logged in user has been changed. The intention for this change was the prevention of blocking licenses by login sessions which were technically active but in fact unused. The two most prominent cases in which unused login sessions were kept active and consumed a license until session expiration have been addressed.

This could have happened, if the user closed his browser without logging out from the CM6 Web Client. In such a case the active login session and thus the license now is reused when starting the browser again. Previously this required a new login and a second license until the first session timed out. Please be aware that this will only work if the same browser is used and cookies are enabled for the browser.

The other case addressed is in a clustered server environment with two or more server nodes. If the server node a user is connected with becomes unavailable and therefore the user is redirected to another node, the user must log in again. However, now the session and license of the unavailable node are reused on the other server node. This redirection mechanism requires a load balancer as single access point to the different nodes. Earlier this created a new session and consumed a second license until the other login session expired.

2.3.5 Cluster capability for New Incoming Mail Handler (NIMH, #626270, #626233)

The New Incoming Mail Handler (NIMH) has been extended so that it now can safely be used on several nodes in clustered environments. The functionality has been extended so that the different nodes can access mailboxes in a co-ordinated manner. For a safe operation in clustered environments the property `mailbox.polling.threads.mail.log.enabled` in the module `cmas-nimh` must be set to *true*, which is the property's default value. Without this setting there is the chance that incoming e-mails may be processed several times by different cluster nodes which will lead to unwanted results. The JBoss cluster will propagate the setting across it nodes after the property has been set. The functionality enabled by this property allows to co-ordinate the cluster nodes' mail processing.

2.3.6 Method for extracting a mailbox URI in NIMH (#626639)

The MailSupportService interface of NIMH did not provide a way to extract a mailbox URI in mail processing scripts. To be able to achieve this in a single method call the new method extractMailIncomingMailboxURI has been added:

```
extractMailIncomingMailboxURI
String extractMailIncomingMailboxURI(MailHolder pMessage,
                                     MailPipeContext pMailPipeContext)
Extracts URI of the mailbox which received the mail e.g
pop3s://test@pop.gmail.com:995.
Uri does not contain the password.
Parameters:
pMessage - the message
Returns:
URI of the mailbox which received the mail
```

2.3.7 Unnecessary indices removed for MySQL (#624809)

Several unnecessary duplicate indices were removed from the data warehouse (CMRF) database schema for MySQL databases. This should result in performance improvements for updates in this context.

2.3.8 Thread identification in log files on cluster environments (#625851)

In cluster environment the JobExecutor threads could not be associated to the cluster node they were running on in the logs. This has been changed and the thread names are now is prefixed with node ID and suffixed with thread index so that now the node can be identified in the log files.

2.3.9 Retry interval timer unit changed for consistency (#626609)

The property "timerRetryInterval.seconds" for the jobExecutor in Admin Tool used Milliseconds instead of Seconds. This inconsistency has been fixed and now seconds are used.

2.4 Known Issues

2.4.1 Changed Configuration of operationtimes.log configuration on JBoss 7

In JBoss 7 installations the configuration for activating performance time measurements with `operationtimes.log` changed.

However, there was a bug present in all 6.9.3 versions of CM6 prior to release 6.9.3.8 which made this time logging dysfunctional. It is fixed for the 6.9.4 releases, so `operationtimes.log` can be used again!

The addition to logging configuration has to be made in the central configuration file `cm6.xml`, or `cm6-cmrf.xml` respectively, and it must be put in the section within the tag `<subsystem xmlns="urn:jboss:domain:logging:1.3">`. Since this configuration is done outside of Log4J, a server restart is required for a logging configuration change to become active. This is the addition required:

```
<size-rotating-file-handler name="OPERATION_TIMES" autoflush="true">
  <file relative-to="jboss.server.log.dir" path="operationtimes.log"/>
  <append value="true"/>
  <rotate-size value="300m"/>
  <max-backup-index value="6"/>
  <formatter>
    <pattern-formatter pattern="%m%n"/>
  </formatter>
</size-rotating-file-handler>

<logger
category="com.consol.cmweb.client.webapp.timemeasure.log.Log4jOperationL
ogger">
    <level name="DEBUG"/>
  <handlers>
    <handler name="OPERATION_TIMES"/>
  </handlers>
</logger>
```

This configuration will be directly available, however commented out, in an upcoming release.

2.5 Bugs fixed

Number	Description
622560	On the detail search page an engineer was able to select, add, or remove result table columns associated with queues he has no access to. This problem was resolved and this kind of column is not available for the engineer anymore.
622586	The web customization "removeContentOnTabSwitch" did not work properly when it was set to "true" in the context "acimSection/ticketEditPage". It worked in the context "acimSection". This problem was fixed and the web customization now works in both contexts as expected.
624102	An error occurred when trying to export template which contained images. The export failed for this reason. This issue has been resolved and export of templates with image is possible again.
625179	In case two different engineers opened the same ticket for editing and tried to save without changes at the same time, for one of the engineers the edit mode kept showing. This specific error was corrected and for both engineers edit mode should end now.
625183	Importing a new system configuration which contains changed workflows including scope name changes could cause errors during import. This problem has been fixed and this kind of import error does not occur anymore.
625240	A Javascript error occurred when switching between the comment, e-mail, attachment tabs in Internet Explorer 8. The error did not impair the web client usage. The issue has been fixed and the error does not occur anymore.
625424, 626024, 626034	When pasting some text from the clipboard into the editor while writing a comment or an e-mail line breaks were wrongly added around the pasted text. This unwanted behavior has been changed so that no line breaks are added around pasted text anymore.
625592	Line breaks entered in CM/Track did not show in the web client and Cm/Track did not show text formatting entered in the web client. These deficits in displaying text have been addressed and now the display should be as expected on both sides.
625625	Adding an attachment along with a new comment in CM/Track associated the attachment to the previous comment and not to the newly added one. This undesired behavior has been corrected and the attachment is now associated with the new comment.
625659	In case a detail search was saved in the favorites section while a specific customer group was generally selected, an error occurred, if this search was executed later from the favorites while another customer group was selected. This error has been remedied and under these circumstances now a message is shown informing the user that the customer group from the search parameters is unavailable in the current context.
625675	The previous version of the Admin-Tool allowed to remove the main administrator e-mail address which resulted in an exception to be presented to the user. This undesired behavior has been changed and now the previous entry is restored, if the field is cleared without providing a new value for it.
625778	An export of templates only in Admin-Tool failed and subsequently caused different exports to fail. The problem has been corrected and exports in the Admin-Tool should not fail anymore, not even for templates only.
625804	A ticket could not be loaded due to an exception when a ticket attachment had been removed before and this removal had caused a defective history entry. This erratic behavior has been corrected and such tickets can be loaded and displayed again.
625842	Creating a ticket caused an exception when the start node of a workflow was inside a scope with timers. This problem was resolved and the exception should not appear anymore.
626023	A supported locale could not be deleted from the system because of a timeout, if it contained too many values. The limit beyond which this problem arose was at several ten thousands of entries in the locale. This issue has been corrected and the timeout does not occur anymore. Now even locales with tens of thousands of entries can quickly be deleted.
626028	After correcting incorrect mailbox connection configurations for NIMH unwanted exceptions were logged in the log files. This type of undesired logging has been corrected and these exceptions will not show in the logs anymore after correcting NIMH configurations.

626054	A fact table could wrongly be removed from the data warehouse (CMRF) database schema during an update, if the corresponding custom field group was removed (and maybe readded) in the Admin-Tool. This wrong deletion was fixed and now the fact table will not be deleted when removing a custom field group.
626108	Scene import failed with an IllegalState exception when the import flag "Delete existing data" was set and an attachment had been removed right before export of the scene. This issue was fixed and import succeeds in this case now.
626114	The index.log file filled up with irrelevant warnings regarding connections on the JBoss 7 platform. This weakness has been corrected and the warnings will not be logged anymore.
626366	If an e-mail template included an image, the outgoing mail did not contain this image even though it was displayed in the e-mail editor of CM6. This error has been fixed and the outgoing mail now includes the image from the template.
626390	Activated time triggers of type repeatable in a workflow which were inactive did not get removed from the active triggers table. This caused the job executor to hang. This error has been corrected and now in such cases a corresponding log entry is written and the trigger in question is deactivated.
626408	Configuration changes in the Admin-Tool could cause an unnecessary high number of indexer tasks. For some setups these could not even be executed. This problem was addressed and the indexer tasks are reduced to the necessary and will be executed.
626500	A missing method invocation could cause the scheduler to not terminate properly when shutting down the system so that the system termination had to be forced externally. This has been corrected and the system should shut down properly without the need to force this externally.
626534	When entering formatted text in CM/Track sections marked as "Heading 1" or "Heading 2" were changed to some other format with blue text color which was rendered differently in the web client. This undesired format change was fixed and now these headings are displayed consistently no matter if they have been entered in CM/Track or the web client.
626540	In CM/Track images which are wider than the display box container were just flowing over the container edge instead of only being shown within the container using a scrollbar. This display error has been corrected and wide images are only shown inside the box using a horizontal scrollbar now.
626541	In CM/Track tables entered in the web client were rendered completely unrelated to the display box container structure of CM/Track which impaired display and usage with this type of entry. This display problem has been resolved and tables are now shown properly inside the display box container.
626564	When upgrading an installation to release 6.9.4.0 the upgrade could fail, if the database engine used was MySQL and specific tables used the MyISAM storage engine due to wrong database system configuration. This problem was fixed and for all tables the correct storage engine InnoDB is used now. It will be used in new installations and the upgrade now fixes this in existing installations, if these used the wrong storage engine.
626600	Time triggers sporadically were not executed during system startup when their defined escalation time was exactly in the workflow engine initialization phase. This error was corrected and time triggers are generally executed even for the workflow engine initialization phase.
626608	The SecurityProvider did not return the correct engineer locale when queried within onAuthenticationFail. This error has been fixed and the correct locale is returned now in this specific case. The SecurityProvider did not allow to provide a custom message to return from onAuthenticationFail. This missing option is provided and now the message to be returned can be specified. The user session in the database was not closed correctly by CmWebSession after SecurityPolicy returned abort as result. This issue has been corrected and the session is terminated correctly in the database entry as well.
626706	The data warehouse LIVE mode could potentially cause a constraint conflict requiring a manual update of the data warehouse on MS-SQL Server. Due to the conflict this update also failed. To prevent this conflict a new constraint has been added so that this problem will not arise in the future.

626790	An InfiniteWorkflowLoopException could occur sporadically when setting/unsetting autocomplete ENUM customs fields, potentially causing memory leaks and business logic errors. This problem has been corrected and the exceptions do not occur anymore.

3 Version 6.9.4.2 (31.03.2015)

Version 6.9.4.2 includes 6.9.3 versions up to 6.9.3.9, 6.9.2 versions up to 6.9.2.11, 6.8 versions up to 6.8.5.8 and 6.7 versions up to 6.7.13

3.1 Update and installation instructions

The default value for the property *mail.on.error* (module *cmas-nimh-extension*) has been changed from "false" to "true". The value will be automatically set to "true" with this update, so please be aware that you may have to disable these mail notifications again in case you intentionally have set the property value to "false". For more details see section 3.3.1.
Please note this is a change in default behavior for the NIMH component!

The dashboard features new functionality to print charts. This function is enabled by default, and it must be explicitly disabled if this should not be possible. For more details see section 3.2.4.
Please note this is new behavior enabled by default!

No further instructions available.

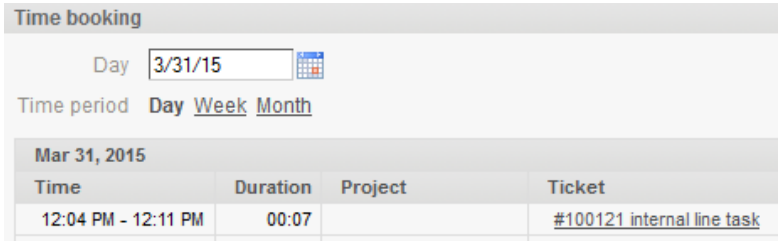
3.2 New Features

3.2.1 Automatic time booking: Tracking of editor usage (#626987, #627271)

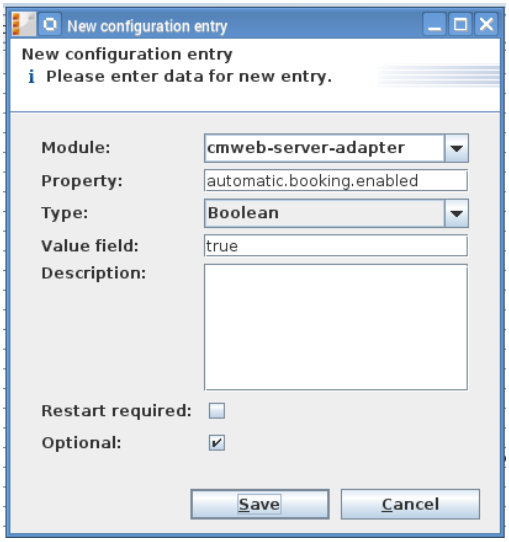
A new feature has been introduced in CM6 regarding the working time documentation: Automatic tracking and booking for usage of the text editor. All additions to the history will be tracked from start (accessing the sections editing functionality) to submission and this duration will be added as a time booking. The same applies when creating a ticket. In this case opening the page marks the start and ticket creation is used for the ending of the interval. Canceling the respective operation or closing the page will not create a time booking. A booking will also not be created when logging out or the session times out. The timer will be suspended when a ticket is saved to the workspace and a different page is being accessed. The minimum time to be recorded is one minute and each duration recorded is rounded up to the minute.

This functionality can be used in conjunction with the new trigger event described in the following section to provide information in a dedicated custom field about the total time spent on the current ticket.

The booking itself will record the duration as well as the start and end time, the date and the ticket. It does not save a project, please see the example booking below. It shows a booking created automatically in the listing on the engineer profile page.

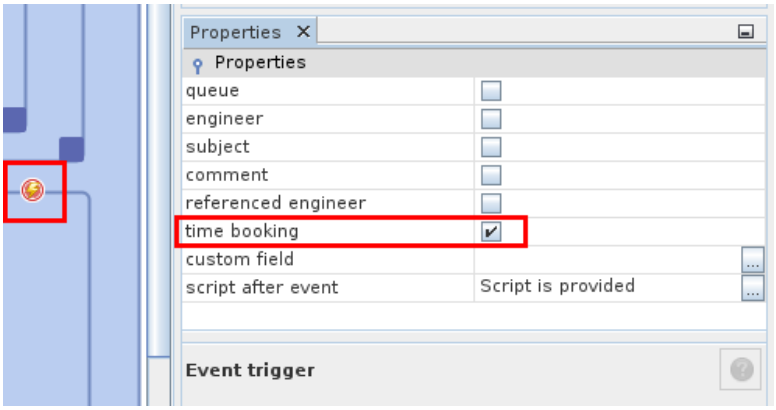


The new automatic time booking feature must be explicitly enabled. It is disabled on newly installed and updated systems. If the property is not automatically added during an update, the new configuration property must be created in the Admin-Tool in the module *cmweb-server-apapter*. The property must be named *automatic.booking.enabled* and it must be of the type *Boolean*. Its value must be set to *true*. After defining this property setting the bookings are made automatically in the way described above.



3.2.2 Automatic time booking: New event for workflow trigger (#626988, #627179)

A new property for event triggers in workflows has been added. This property when set causes the trigger to fire whenever a time booking is entered. A script supplied for such a trigger can for example add up all the previous time bookings for this ticket and display the total sum in a dedicated custom field.



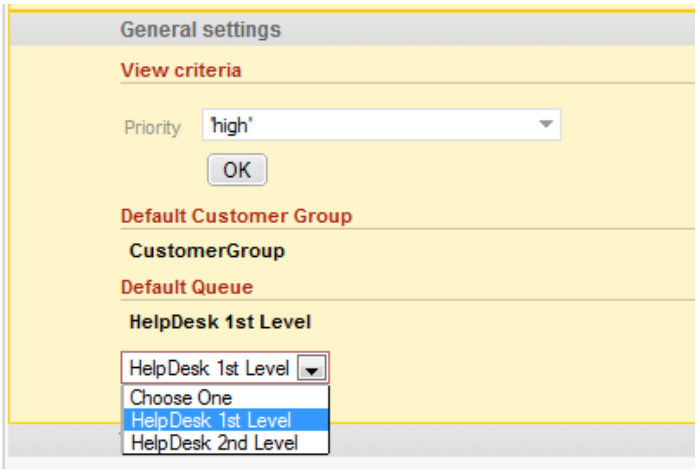
For event triggers in the process designer there is a new property labeled “time booking”. This property when checked causes the trigger to fire whenever a time booking is submitted. A script provided would then be executed. This applies to both manual time bookings and automatic ones as described in the section above.

An example script – which relies on a specific system configuration with the named custom fields – is shown below. It uses an array to keep the sums of bookings per engineer and adds the booking just made to the current engineer’s bookings.

```
import com.consol.cmas.common.model.time.TimeBooking;

long bookedMinutes =
((TimeBooking)workflowApi.getEvent().getSource()).getTimePeriod().getBookedTime()/60/1000;
String currentEngineer = workflowApi.getCurrentEngineer().getName()

List bookings = ticket.get("time_booking.bookings")

int numberOfBookings = 0

if (bookings != null){
  numberOfBookings = bookings.size()
}

for (int i = 0; i < numberOfBookings; i++){
  if (ticket.get("time_booking.bookings[" + i +
"].engineer").equalsIgnoreCase(currentEngineer)){
    ticket.set("time_booking.bookings[" + i + "].time",
               ticket.get("time_booking.bookings[" + i + "].time") + bookedMinutes)
    return;
  }
}
// in case no engineer was found in the bookings add new entry for her/him
ticket.add("time_booking.bookings", new Struct().set("engineer",
currentEngineer).set("time", bookedMinutes))
```

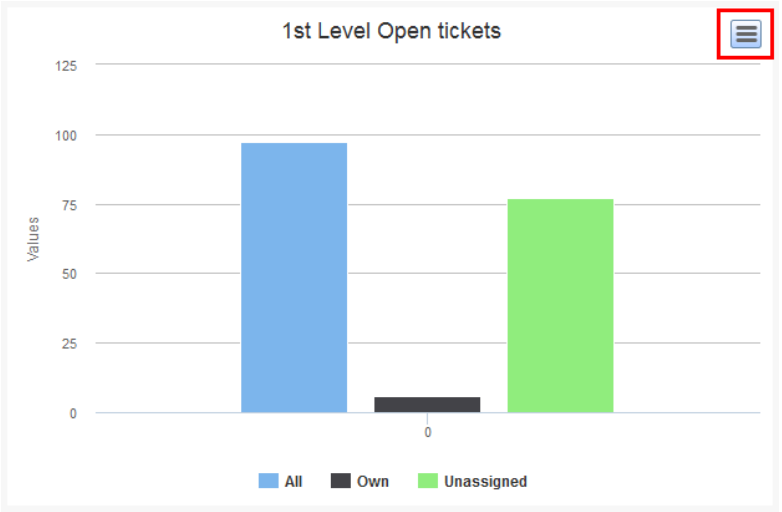
3.2.3 Default queue setting in engineer profile (#611996, #627198)

A new configuration setting for each engineer has been introduced: the definition of the default queue. The queue defined as default will be preselected when creating a new ticket. A drop-down menu selection has been added to the engineer profile page for this setting. In this menu the queues are listed for which the engineer has “create ticket” permissions. Disabled queues will not be listed. After a setting has been made the selected default queue is shown above the drop-down menu for selection which can be seen in the screenshot below. If there is only one selection possible only the queue is shown and the selection is omitted. In case the required permission of the engineer is revoked the selection will also be removed.



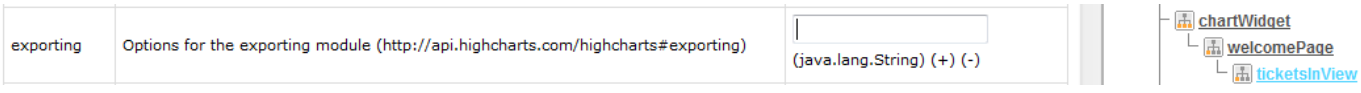
3.2.4 Extensions of the Dashboard chart functionality (#626944, #626894)

The charting functionality of the dashboard has been enhanced by adding extensions to the library used. The new functionality includes chart printing, data drilldown and 3D rendering. Please note that the printing functionality is activated by default. It must be explicitly deactivated, if desired.



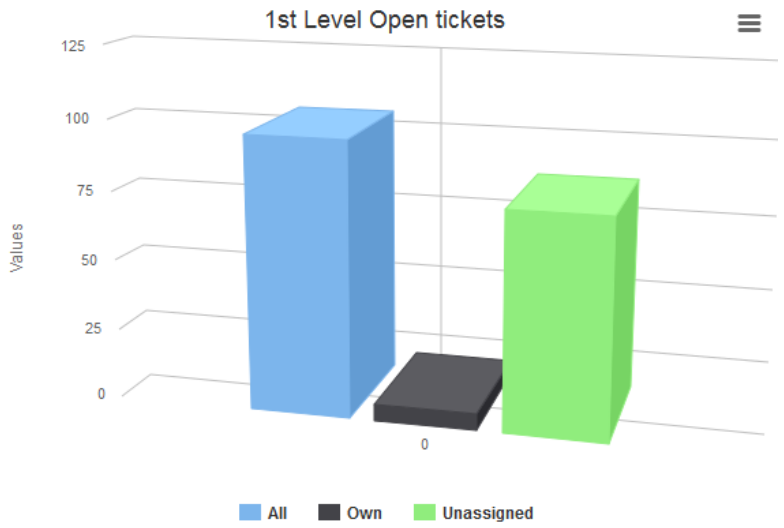
Each chart now displays a button in the upper right corner as highlighted in the screenshot. Clicking this button will print the chart using the system print dialog. This can be explicitly disabled in the page customization for this chart using the attribute “exporting” and setting it to the value `enabled: false`. The behavior can be configured by this page customization using the API described here:

http://api.highcharts.com/highcharts#exporting .



The default values and the values set via the page customization can be overridden by the return value of the data script.
In principle it is possible to use this functionality for export of images of the chart, but this requires an external export server. For this reason the functionality is not recommended and disabled by default.

Another extension has been added for the charting component of the dashboard. It allows for 3D rendering of the charts as illustrated by the screenshot.



This kind of 3D rendering for the data can either be configured by setting the additional options in the chart's page customization or in the return value of the data script. Using the page customization attribute "chart" 3D options are defined as JSON objects. The available options are documented in the highcharts API documentation on the website:

http://api.highcharts.com/highcharts#chart.options3d

For more information on the 3D implementation and concepts of the solution see

http://www.highcharts.com/docs/chart-concepts/3d-charts

Caution! Usage of this kind of rendering puts a rather heavy performance load on the browser. Please be sure that the client environments are capable of displaying these 3D charts when implementing them!

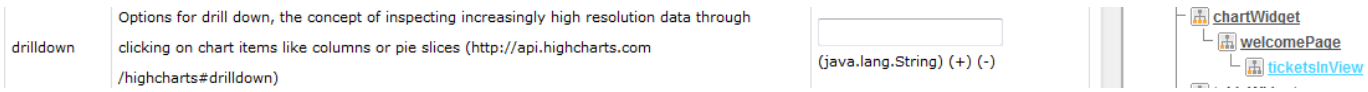
An example script for showing the default chart in a 3D view is listed below. It sets the 3D options in the return value of the script. The additional options in comparison to the standard script are highlighted.

```
import com.consol.cmas.common.model.ticket.*;
import com.consol.cmas.common.model.ticket.view.*;
import java.util.*;
import java.util.Map.Entry;

if (viewId == -1) {
```

```
    return [visible: 'false']
}
def engineerLocale = engineerService.getCurrentLocale()
def view = viewService.getById(viewId)
def viewName = localizationService.getLocalizedProperty(View.class, "name", viewId,
engineerLocale)

ViewCriteria allCriteria = new ViewCriteria(view,
        ViewAssignmentParameter.allTickets(),
        ViewGroupParameter.allTickets(),
        new ViewOrderParameter())
def allTickets = ticketService.getIdsByView(allCriteria)

ViewCriteria ownCriteria = new ViewCriteria(view,
        ViewAssignmentParameter.allTickets(engineerService.getCurrent()),
        ViewGroupParameter.onlyOwnTickets(),
        new ViewOrderParameter())
def ownTickets = ticketService.getIdsByView(ownCriteria)

ViewCriteria unassignedCriteria = new ViewCriteria(view,
        ViewAssignmentParameter.allUnassignedTickets(),
        ViewGroupParameter.onlyUnassignedTickets(),
        new ViewOrderParameter())
def unassignedTickets = ticketService.getIdsByView(unassignedCriteria)

def data = []
data.add("{name: _('all'), data:[${allTickets.size()}]}" as String)
data.add("{name: _('own'), data:[${ownTickets.size()}]}" as String)
data.add("{name: _('unassigned'), data:[${unassignedTickets.size()}]}" as String)


return [series: "[${data.join(',')}]" as String,
    visible: 'true',
    chart: "{type: 'column',
             options3d: {enabled: 'true', alpha: '15', beta: '15', depth: '50',
                         viewDistance: '25'}}",
    plotOptions: "{column: {depth: '25'}}",
    title: "{text: '${viewName}'}" as String,
    tooltip:"{headerFormat:''}" ,
    localization: "de: {all:'Alle', own:'Eigene', unassigned:'Nicht zugewiesene'},"
                + "en: {all:'All',  own:'Own', unassigned: 'Unassigned'}"];
```

The last extension added for this release enables basic drilldown functionality for charts in the dashboard. The page customization attribute “drilldown” can be used for general settings.

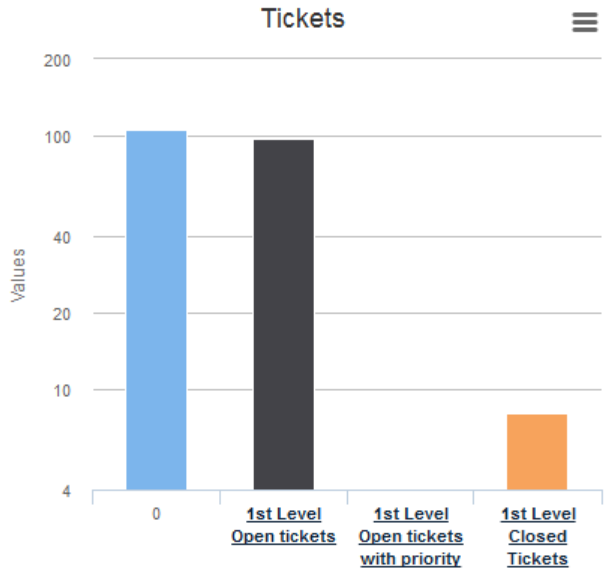
http://api.highcharts.com/highcharts#drilldown

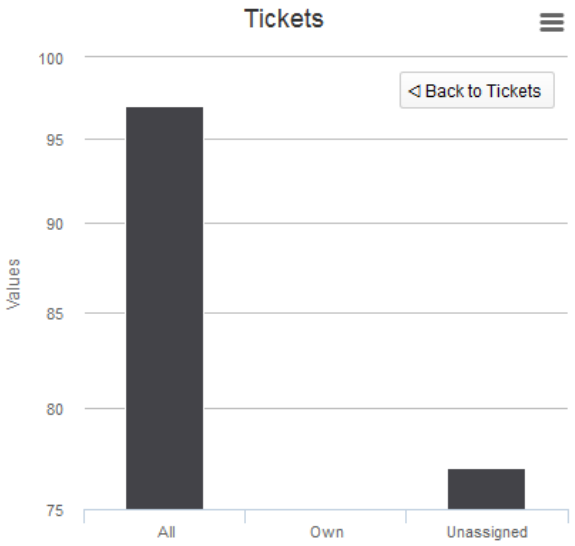


However, to be reasonably used by this functionality the data script should be extended, too. The data in the return script needs to be extended with the data shown in the drilldown. These additional data must be referenced with the data they extend. For a description of the concepts see the highcharts documentation:

http://www.highcharts.com/docs/chart-concepts/drilldown

The effect is that a second level of data can be shown in the same chart when clicking on a subset. In the left screenshot the columns are clickable and the column name labels are links. After clicking on either a detail view of this subset is shown, illustrated by the right screenshot.





The color of the columns in the detail view is the color of the selected subset in the main view. The detail view also displays a back button on the upper left side of the chart. The following data script dynamically provides the data for this drilldown.

```
import com.consol.cmas.common.model.ticket.*;
import com.consol.cmas.common.model.ticket.view.*;
import java.util.*;
import java.util.Map.Entry;


if (viewId == -1) {
    return [visible: 'false']
}

def engineerLocale = engineerService.getCurrentLocale()

def views = []

def seriesdata = []
def drilldownseries = []
def allTicketsCounter = 0
views = viewService.getByEngineer(engineerService.getCurrent())

for (view in views){
    def viewName = localizationService.getLocalizedProperty(View.class,
          "name",
          view.getId(),
          engineerLocale)

    ViewCriteria allCriteria = new ViewCriteria(view,
          ViewAssignmentParameter.allTickets(),
          ViewGroupParameter.allTickets(),
          new ViewOrderParameter())
    def allTickets = ticketService.getIdsByView(allCriteria)
```

```
    ViewCriteria ownCriteria = new ViewCriteria(view,
          ViewAssignmentParameter.allTickets(engineerService.getCurrent()),
          ViewGroupParameter.onlyOwnTickets(),
          new ViewOrderParameter())
    def ownTickets = ticketService.getIdsByView(ownCriteria)

    ViewCriteria unassignedCriteria = new ViewCriteria(view,
          ViewAssignmentParameter.allUnassignedTickets(),
          ViewGroupParameter.onlyUnassignedTickets(),
          new ViewOrderParameter())
    def unassignedTickets = ticketService.getIdsByView(unassignedCriteria)

    seriesdata.add("{name: '${viewName}',
          y: ${allTickets.size()},
          drilldown: '${view.getName()}'}")


    def data = []
    data.add("['All', ${allTickets.size()}]")
    data.add("['Own', ${ownTickets.size()}]")
    data.add("['Unassigned', ${unassignedTickets.size()}]")

    drilldownseries.add("{id: '${view.getName()}',
          data:[${data.join(',')}]}" as String)

    allTicketsCounter += allTickets.size()
}

return [series: "[{name: 'Tickets', colorByPoint: true,
          data: [${allTicketsCounter}, ${seriesdata.join(',')}]}]" as String,
        drilldown: "{series: [${drilldownseries.join(',')}]}" as String,
        visible: 'true',
        chart: "{type: 'column'}",
        title: "{text: 'Tickets'}",
        xAxis: "{type: 'category'}",
        yAxis: "{type: 'linear'}",
        legend: "{enabled: false}"
];
```

3.3 Changes

3.3.1 NIMH property "mail.on.error" default value changed (#627283)

The default value for the property *mail.on.error* (module *cmas-nimh-extension*) has been changed from "false" to "true". This restores the same default behavior as earlier MULE mail handling, however, it changes the previous default behavior of the NIMH component.
Please note this is a change in default behavior for the NIMH component!

The value will be automatically set to “true” with this update, so please be aware that you may have to disable these mail notifications again in case you intentionally have set the property value to “false”.

3.3.2 Modified order NIMH fetches e-mails from the database (#627274)

The order the NIMH component fetches e-mails to be sent from the database has been modified. Now e-mails with errors registered and the oldest queued e-mails have precedence. This helps ensure that notification e-mails will not be queued for a long time without being sent.

3.3.3 Logging for pasting of images improved (#625644)

The logging for pasting of images was improved. It can occur that images pasted together with HTML text cannot be displayed anymore later on. They are not included but linked and the link may not be accessible anymore. This can now be better traced. In a future release improved handling of this kind of image pasting will be implemented. This is only a partial solution for the original problem. Please refer to the section “Known issues” as well!

3.3.4 Example configuration for "operationtimes".log added (#626389)

A working example configuration of the performance information logging with an "operationtimes.log" file has been added. The configuration is added in the correct configuration file for each application server product, but the section is commented out, so that this logging is turned off by default. This way it can be easily activated and configured.

3.3.5 Timeout when trying to open template administration (#626284)

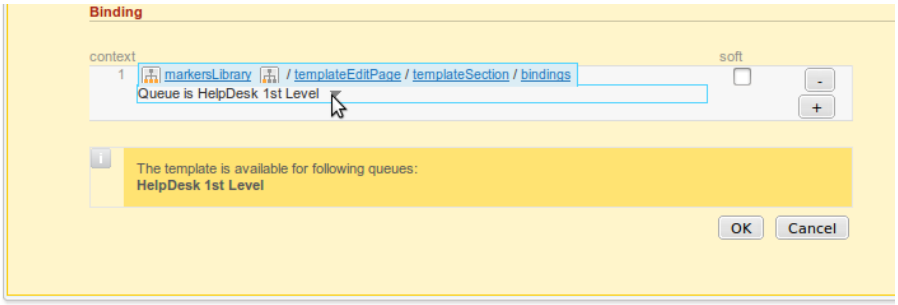
A timeout could occur when tying to access the template administration, if the system is configured with very large ENUMS (several thousand entries). Such a configuration also has a significant general performance impact. To be able to use the template administration in such a case two enhancements of the system have been implemented: localization cache warmup on the server side for more efficient retrieval of the values and a page customization on the Web Client side to be able to avoid problematic ENUM field binding of templates.

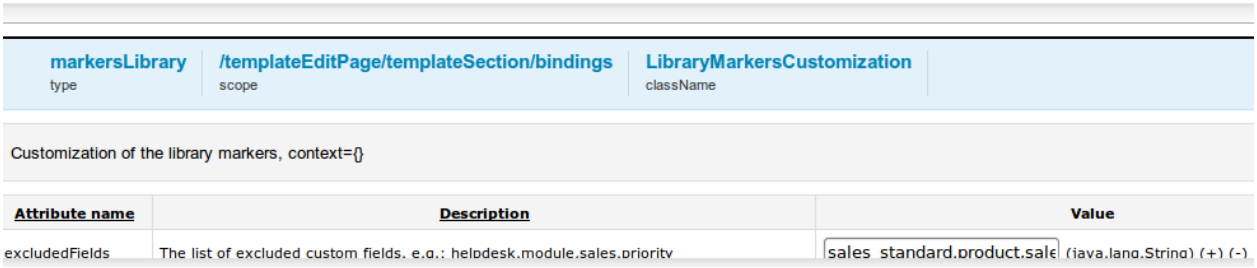
The localization cache warmup is configured by properties in the Admin-Tool. The properties are located in the module *cmas-core-server*. The cache warmup is activated by default with the property *warmup.executor.enabled* set to “true”, so it can be disabled in case of issues.

The cache warmup can be executed via JMX bean *consol.cmas:type=config, topic=global, name=serverWarmupExecutor*, and it can be extended easily via the ServerWarmupExecutorCommand interface.

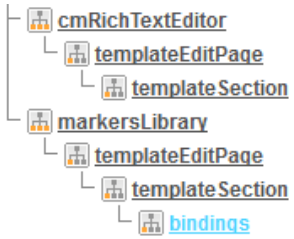
Additionally there is a new page customization which makes it possible to exclude specific custom fields from binding to templates. The problematic kind of fields with many entries is usually irrelevant for binding, so this optimization is possible.

The customization is done on the template editing page specifically for the bindings as the screenshot shows. The attribute is called “excludedFields” and its value is a list of custom field as explained in the description: <GROUP_NAME>.<FIELD_NAME>:





The customization can be found at the lower end of the navigation tree:



3.4 Bugs fixed

Number	Description
624675	When the page customization "removeContentOnTabSwitch" was set to "true" switching from a comment tab to the attachment tab and then to the e-mail tab caused undesired behavior: Text entered as comment was wrongly merged with an e-mail template instead of being discarded. This issue has been resolved and the comment text is dropped as desired.
625226	CM/Phone displayed the caller's phone number for outgoing calls instead of the number called when used with some specific telephony systems. This problem has been addressed and should not show any more for the affected systems.
625777	When deleting a ticket relation and then undoing this, the customer name field is cleared with the label being shown, but its value is displayed empty. This issue has been resolved and the name is displayed correctly after this operation.
625932	Striking variations in the font size of an e-mail could be observed comparing the display in the CM6 Web Client and Outlook e-mail client. Sizes were too small in CM6. For this reason the font size unit has been changed from px to pt. Some minor issues remain and will be fixed in a future release.
626214	It was not possible to enable page customizations in version 6.9.4.0 or later when using Internet Explorer 8. This has been corrected and page customizations can be enabled in all supported browsers now.
626640	When editing customer data it could happen under some circumstances that the CM/Track user role associated with this customer for the CM/Track login was removed. This problem has been fixed and the association will not be deleted anymore.
626690	Updates of text class information to new tickets sent to the data warehouse could cause an exception and terminate the DWH live mode. This occurred mainly when using JMS messaging and switching to DIRECT mode could be a workaround. The problem has been fixed and DWH live mode is not terminated by such an exception anymore.
626763	The system property *unitIndexSearchResultSizeLimit* (module *cmweb-server-adapter*) did not apply for customer suggestions. This error has been corrected and now it applies for these, too.
626889	An IllegalStateException in the context of login was logged when using Internet Explorer versions 8 and 10. Other browsers were not affected. The problem was connected with the way favicons were requested and it has been resolved.
626942	The label displayed for incoming mail was using a technical text instead of the language specific text when using NIMH. This has been corrected and now the desired label is shown for incoming mail when using NIMH.
627002	A constraint violation while deleting data could occur when importing a scene with static configuration data. This error has been corrected and such an import is working again.
627085	The dashboard chart was not displayed, if the first value of the data series was zero. This error has been corrected and charts are now rendered even when the first data value is zero.
627098	In the latest version 6.9.4.1 there was a unique constraint violation error in the update scripts regarding some page customizations, mainly relating to sections on the ticket page. A workaround has been available, and this problem has been fixed generally now, Thus, updates with these page customizations will work.
627174	A wrong sender name for e-mails just sent could be displayed in the history entry, if the same e-mail address was used by several engineers. This has been corrected and now the correct name is shown.
627214	In specific cases empty lines were removed from a comment or e-mail after submitting it when based on a template. The appearance of the message was impaired then since this is used to structure and layout the text. This problem has been resolved and newlines are now preserved.

627243	The calendar icon to access the date chooser for a date field disappeared after changing the selected queue while creating a new ticket. This unwanted behavior has been corrected and the icon will not disappear anymore.
627266	The extension which allows to define dependent read-only fields allowed unwanted editing of the fields under some circumstances. Editing should be prohibited depending on the value of another field. Though after legitimately editing some other entry a field that should have been read-only could be edited. This undesired behavior has been changed and dependent fields should be read-only now when desired.

3.5 Known Issues

Number	Description
625644, 627286	Pasting of images linked within some pasted HTML text still relies on the external source which may not be available at a later time or for other users. A proposed solution will be implemented in a future release.
626984	The message that the session has expired is not displayed on the login page, if after session expiry the browser's reload button is clicked instead of a link.
627234	When using the Internet Explorer browser version 9 text lines can be displayed too close to each other in the comment/e-mail editor. If the font size is 14pt or bigger the characters on adjacent lines overlap obscuring each other.

4 Version 6.9.4.3 (05.05.2015)

Version 6.9.4.3 includes 6.9.3 versions up to 6.9.3.9, 6.9.2 versions up to 6.9.2.11, 6.8 versions up to 6.8.5.8 and 6.7 versions up to 6.7.13

4.1 Update and installation instructions

4.1.1 Support for WildFly 8.2.0 Application Server Platform (#627033)

ConSol*CM now supports the application server platform WildFly (version 8.2.0 Final) starting with this release 6.9.4.3. This product serves as basis for current versions of the JBoss application server platform, so most information about the latter applies in this context, too. JBoss version 7 (JBoss EAP 6.2.0GA) are extended subscription products based on WildFly 8. WildFly is available freely without subscription from this site:

http://wildfly.org/

ConSol*CM System requirements for use with WildFly 8.2.0:

Operating systems:

- Linux with a kernel 2.6.24 or newer or
- Windows Server 2008 and 2012, in case of limited performance requirements like for specific test systems client versions as well, namely Windows Vista or newer (64 Bit support starting with Windows 7).

Java runtime version:

- Java 1.7.0 Update 51 or later

Hardware (recommendation for a typical installation of 50 concurrent users):

- at least 10 GB of memory (RAM)
- at least 10 GB of hard drive storage
- 2 GHz Dual Core processor or equivalent.

ConSol*CM - Available Distributions for WildFly 8.2.0:

- For Oracle: dist-package-distribution-6.9.4.3-oracle-wildfly-8.2.zip
- For Microsoft SQL Server: dist-package-distribution-6.9.4.3-mssql-wildfly-8.2.zip
- For MySQL: dist-package-distribution-6.9.4.3-mysql-wildfly-8.2.zip

Update

WildFly support starts with ConSol*CM version 6.9.4.3. Thus, the system must be newly deployed on an initially new WildFly installation. The update option will be available for the following releases.

CM6 and CMRF Migration

WildFly 8.2.0 and JBoss EAP 6.2.0GA installation are identical in most aspects. They differ only in very few details. For this reason only the relevant differences are explained in this section. All other aspects and details of a migration of ConSol*CM to the WildFly application server platform are documented in the Release Notes of CM version 6.9.3.3.

Please refer to the section about JBoss EAP 6.2.0GA support and migration in the ConSol*CM 6.9.3 Release Notes for detailed step-by-step descriptions of the migration procedure!
Please note the other existing documentation regarding JBoss EAP 6.2.0GA, which should address WildFly support and the few existing differences as well.

The following paragraphs only describe the relevant differences of WildFly 8.2.0 setup from the JBoss EAP 6.2.0GA configurations. Due to these (and some more minor) configuration differences it does not work to copy a configuration file from an existing JBoss 6.2.0GA installation to a WildFly installation!

CMRF URL configuration
The CMRF URL setting for the DWH configuration in the Admin-Tool must be a different value for use with WildFly compared to JBoss EAP:
WildFly: http-remoting://<CMRF_HOST_IP>:<HTTP_PORT>
(i.e. http-remoting://192.168.0.1:8080)
For comparison:
JBoss EAP: remote://<CMRF_HOST_IP>:<JNDI_PORT>
(i.e. remote://192.168.0.1:4447)

Clustering domain socket configuration
In a cluster setup the domain socket binding group configuration are used to define the ports which are opened and used for incoming connections. There are fewer components in use for WildFly than for JBoss EAP so there are fewer configuration items. The two socket binding groups “full-sockets” and “full-ha-sockets” are affected by this difference. For each server product a working default/example configuration is provided with the distribution. Thus, for details please see the clustering configuration documentation

Custom CM project dependencies
There is an additional dependency which has to be met when creating custom CM projects. It must be present in the file “jboss-app.xml”. Apart from this the setup corresponds to the one for JBoss EAP. The extra dependency is the following:

```
<module name="org.apache.xalan" export="true"/>
```

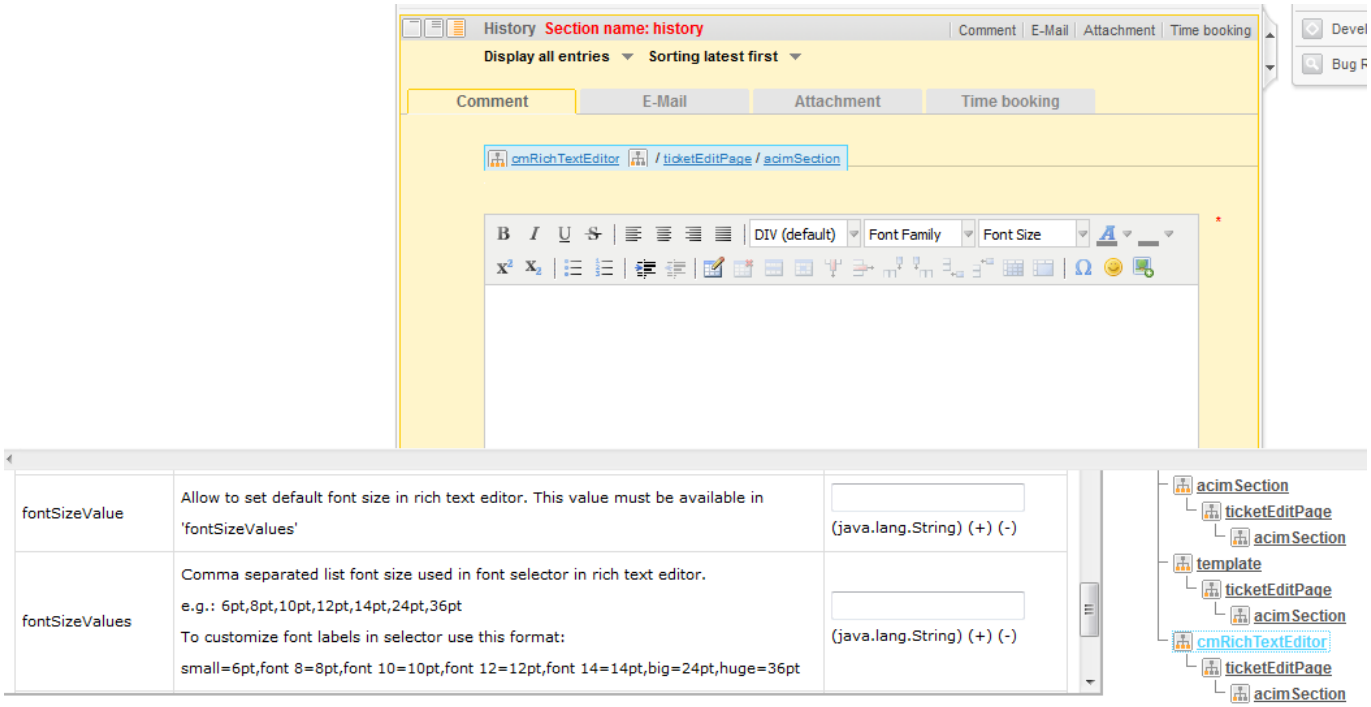
See the custom project documentation of JBoss EAP 6.2.0 and WildFly 8.2.0 for a full account of the configuration.

4.2 Changes

4.2.1 Display Font Size Configuration for Handling Text Sizes in the Rich Text Editor (#625925)

The font size units have changed between 6.8 and 6.9 versions of CM. This partially lead to very small text display sizes in external e-mail clients. In order to deal with this kind of problem which is highly dependent on environment details, two new page customizations have been introduced. These can be used to configure the font sizes in the rich text editor including the units, their presentation and the default size.

The customization attribute names are “fontSizeValue” and “fontSizeValues”. They can be accessed after opening a rich text editor on the ticket edit page for example. Once the editor is visible the page customizations can be enabled and the subtree “cmRichTextEditor” becomes available:



The customizations are available on the ticket edit and template edit pages. The values can be set separately when choosing the respective scope, but a global setting can be used when choosing “cmRichTextEditor” as scope like shown in the screenshot above. In case values are set for both the global scope and a specific page the specific value is used on this page.

fontSizeValue: This is the default size for the text in the rich text editor. It must be one of the values form the list in the other customization “fontSizeValues”.

fontSizeValues: This is the list of values offered in the font size selector of the rich text editor. It is a comma separates list of legitimate font size values including their unit like “6pt,10pt,12pt”. The values can be prepended by a label which is shown in the selector instead of the value itself: “tiny=6pt,regular size=10pt,large=12pt”.

Custom project font size settings

Custom CM projects which have already modifications for the font sizes in the rich text editor must merge their changes into the new “editorInit.js”, if this file has modifications. They should ideally remove all custom files like “tinymce_content_<theme_name>.css”, or “editorInit_<theme_name>.js”, if the editor font sizes were the only changes. This kind of setting should be done in the customization now. This adjustment is optional at the moment, however.

In a future release an even more general solution addressing more aspects of text display is planned, though the current configuration option should not become obsolete.

4.2.2 Boolean Fields Representation in the Web Client - Script Adjustments (#627216)

Boolean fields represented as checkboxes show a different default behavior since version 6.9.4.0 (see section 1.3.1). They save the value *null* when left unchecked while setting *false* in this case previously. This may require adjustments of scripts which used the value *false* for identifying the unchecked state previously. In this section a way how to deal with this and a script helping to identify usages the affected fields are presented.

A Groovy script has been created in the context of this ticket. This script must be run from the Admin-Tool using the Task Execution Framework. The script will log usages of boolean fields in Groovy scripts to the console. This output must be reviewed individually and according actions must be taken manually! The script output may give false positives as well as it may not find some occurrences. This is due to the versatile possibilities of creating references to fields and cannot be fully avoided.

Please request this script by contacting the ConSol*CM Support Team, in case you need to use it for identifying the need of further action.

4.2.3 Improvement of Security Configuration for Jolokia Component (#626501)

The configuration regarding the security of the Jolokia component (remoting JMX-HTTP bridge) has been improved. For the Weblogic application server platform further steps are planned and will be implemented in an upcoming release.

4.3 Bugs fixed

Number	Description
626878	Queue permissions which were introduced by importing an exported scene, from a staging system for example, did not work properly. Only after deactivating and reactivating them the system applied the privileges as intended. This faulty behavior has been corrected and this kind of permissions works immediately after import now.
627234	When using the Internet Explorer browser version 9 text lines could have been displayed too close to each other in the comment/e-mail editor. If the font size was 14pt or bigger the characters on adjacent lines overlapped obscuring each other. This visual impairment has been fixed and lines do not overlap anymore.
627261	Importing a scene which contained a standard e-mail template, while another template was defined as standard in the system already, did finish successfully, but the system then had two standard e-mail templates defined. This state subsequently caused exceptions when using templates and the template administration. This problematic issue has been fixed and now the standard template flag is cleared during the import from the import scene, if the standard is already set in the system. It applies to e-mail and comment standard templates. This clearance is logged and displayed in the Admin-Tool dialog after import.
627290	A previously exported scene could not be imported due to unique constraint violations after the some system modifications were made in the meantime. Besides the modification incompatibilities sort order issues caused the violations. These sort order issues have been fixed and a former export can now be imported, provided the necessary manual adaptations for the system changes in between are carried out.
627333	The update to recent CM versions on MSSQL server using a non-Unicode dialect failed due to a problem in an update script. This problem has been resolved and the update works normally on this platform as well.
627356	The ticket list for adding a time booking on the engineer profile page listed closed tickets, too. It happened for those tickets to which the engineer was assigned as an additional engineer. These undesired listings have been eliminated and now only open tickets are shown in this place.
627428	A necessary data warehouse task has not been created when a custom field was annotated as "reportable". This omission could ultimately have resulted data warehouse inconsistency. This error has been corrected and the necessary task is created, thus avoiding potentially inconsistent data warehouses.
627431	The new system property "automatic.booking.enabled" was not automatically added to the system in the update procedure to the latest CM release. This deficit has been addressed and the property will be added by the update to the current system, if it has not been created manually.

4.4 Known Issues

Number	Description
625644, 627286	Pasting of images linked within some pasted HTML text still relies on the external source which may not be available at a later time or for other users. A proposed solution will be implemented in a future release.
627308, 626984	The message that the session has expired is not displayed on the login page, if after session expiry the browser's reload button is clicked instead of a link.

5 Version 6.9.4.4 (21.07.2015)

5.1 Update and installation instructions

No further instructions available.

5.2 Bugs fixed

Number	Description
626581	start.groovy.task.enabled property is not created during update
627127	Dasboard/Chart: renders "No data" if a multidimensional data array is given
620278	Search API: early detection of not indexed search criteria
627418	Process Designer Event-Trigger type "time booking" lost on export/import
627533	Display font size 'layout' problem
627603	Update to 6.9.4.2 failed for SQLServer2005
623145	Engineer Function gone after Scene Import
627691	TEF task should be re-scheduled on error but is started directly instead
627558	Copy/Paste issue from Word to Comment in CM
627789	Ticket stuck after a second manual timer trigger is executed
627839	Missing Webcustomization for default font-family
627888	Page customization set on Scope level or SubScope level doesn't work
627402	Company gets reactivated via ETL update

6 Version 6.9.4.5 (02.10.2015)

Version 6.9.4.5 includes 6.9.3 versions up to 6.9.3.10, 6.9.2 versions up to 6.9.2.11, 6.8 versions up to 6.8.5.8 and 6.7 versions up to 6.7.13

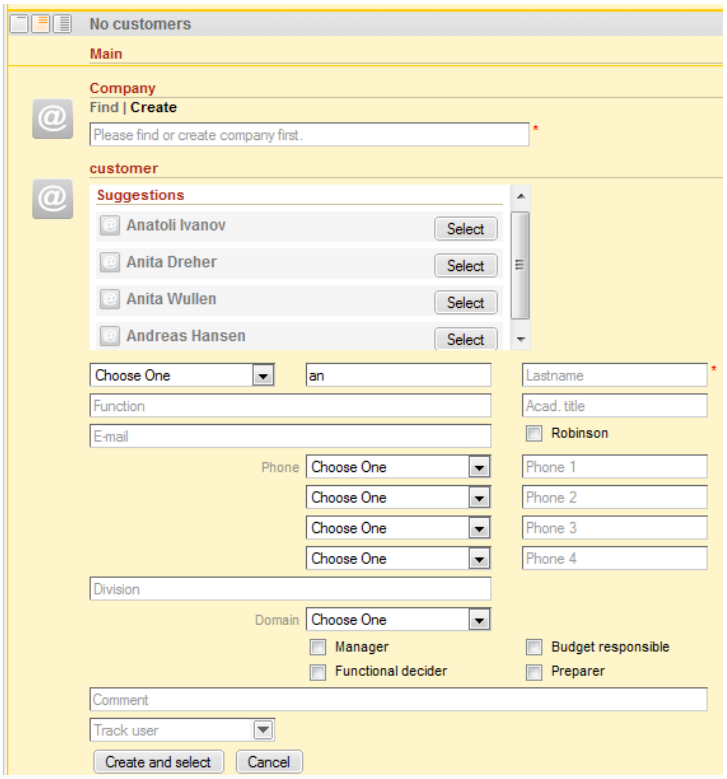
6.1 Update and installation instructions

No further instructions available.

6.2 New Features

6.2.1 Disabling extensive customer suggestions and simple UI for customer selection (#627965)

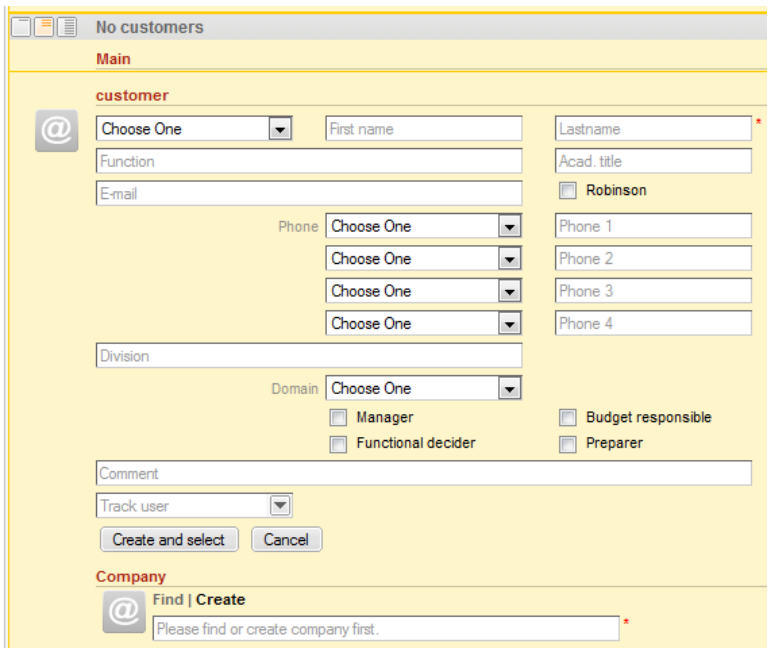
A new web customization has been introduced to provide an alternative customer selection user interface. The choice is made by the value of the new web customization "suggestionsDisabled" for the type "customerSectionPanel" on the ticket create page.



The current default interface presents the contact fields immediately so that it is possible to select a customer based on matching suggestions as well as create a new one directly.

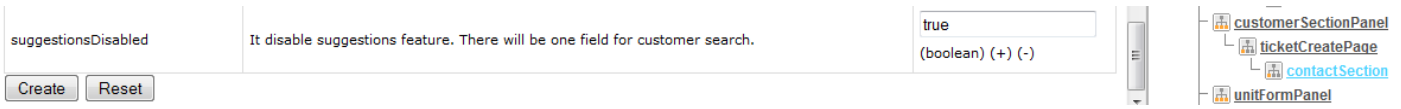
The new alternative mimics the behavior of earlier versions of CM6 in that it offers one single search field. The text entered there is used to match against all relevant contact fields. The matching values are presented for selection. For creation of a new contact the corresponding “Create” link must be clicked.

This allows to the access the form to enter contact data. The same mechanism applies to the company for the new contact. It can be searched with the single search term field, but the creation form for a new company must be opened by the “Create” link.



This alternative contact selection user interface has advantages when contacts are rarely or never added, which is the case, if contacts are retrieved from an external system.

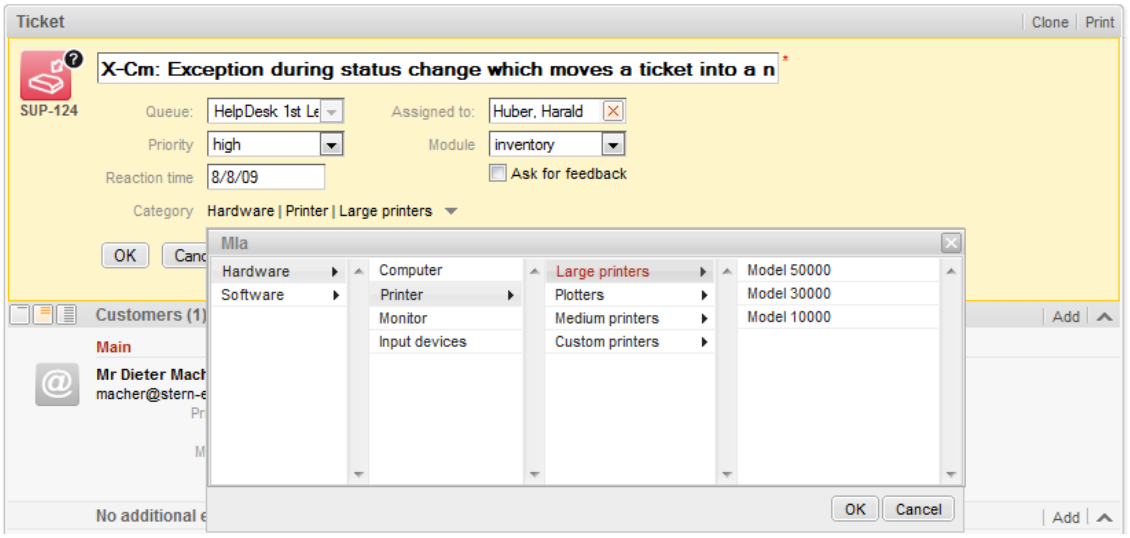
The configuration to achieve this behavior with the single search field rather than the default behavior is controlled by a new page customization. It can be found on the ticket create page and is defined for the type customerSectionPanel. The attribute “suggestionsDisabled” in the scope “ticketCreatePage/contactSection” must be set to the value “true” for the alternative behavior. The value “false” or no setting leads to the default behavior showing the contact fields immediately.



6.3 Changes

6.3.1 Layout improvements

- **Width of the MLA component increased (#627624):** The MLA component in the web client now is wider than before. The default width has been increased by about 20%, so it should be better to use on modern monitors.



6.3.2 Multiple MIME types added to mail processing (#627606, #626322, #626303)

Multiple MIME types for file attachments of incoming mail have been added. These will be recognized correctly now. Mostly office document file formats (like *.docm or *.wkm) have been added, but also PDF types like "application/x-pdf" or "application/acrobat" and SAP files are new. These file types are handled correctly, they are visibly recognized now and will not be rejected any more.

6.3.3 Pasting of images made more consistent (#625644)

The usage of copy and paste in e-mails did not generally yield consistent results. Pasting an image from the ticket history to an external e-Mail-editor may not have worked, occasionally an image sent in an e-mail from CM6 did not show up in the history, or images in e-mails sent by reply or forward from the history were present in this original history entry, but not in the mail. Other cases were that pasting images from an external e-mail program into the CM e-mail editor or pasting images from the CM ticket history into a word document did not work properly. This behavior could vary depending on the browser in use and the usage of the ConSol*CM image paste applet.

This behavior has been reworked extensively to provide a more consistent behavior. The behavior in the CM editor is now as described below **provided that the ConSol*CM image paste applet is activated**!

- Copying an image from the browser into CM editor works as expected when using the keyboard shortcut CTRL-V or the context menu. In Internet Explorer 8 only a dialog “Paste from Word” is shown, and pasting is successful when clicking “Insert” there.
- Copying images from other programs like MS Paint and pasting these works as expected. The same applies for screenshots in the clipboard.
- Copying sections with images from Word also works as expected with the images being shown without problem. There is a difference between browsers concerning which text styles are changed when pasting. Font selection and sizing may be adjusted to the CM editor default fonts and sizes in Firefox and Internet Explorer to a different degree. Coloring and paragraph definitions are kept as in the original document.

Additionally the logging for pasting of images was improved. This can now be better traced in cases unexpected behavior is observed.

6.3.4 Page header meta-tag extension (#628133)

A new x-ua-compatible meta-tag with the setting "IE=edge" has been introduced to the HTML page header, so that some specific compatibility issues for the Internet Explorer browser have been resolved.

6.3.5 Warning messages in log upon failed LDAP login attempts (#626094)

Earlier only an error stack trace was written to the log files upon a failed LDAP login attempt. This has been enhanced by a warning message which informs, if the username entered was not found or if it is not unique, along with logging the username and LDAP id. In a future change the stack trace will be handled as well, see the known issues entry ticket #628191.

6.3.6 Minor performance improvements (#628157, #628107)

Some minor performance improvements have been implemented regarding ticket creation and engineer selection for example.

6.4 Bugs fixed

Number	Description
621674, 625170	Markup exception during ticket display: Occasionally markup errors occurred due to disallowed characters like “<”, “>”, or “&” in Wicket IDs. This error has been corrected, so that the characters are not used anymore and this kind of markup errors should not be present now.
622167	Internet Explorer ticket history printout problem: When printing tickets with long history entries from Internet Explorer, it could happen, that in the printout different history entries were printed on top of each other in the same area. This did not happen with other browsers. The problem has been fixed so that the printouts appear as expected with Internet Explorer as well.
624195	The character "+" was not allowed in e-mail addresses by the validation even though it can be part of a valid address. A data object field needed to be annotated with "validation:email" for this kind of problem ot occur. This was changed so that e-mail addresses with a "+" character are allowed now.
625397	The e-mail validation error message was wrongly shown when a fragment of an e-mail address has been entered for a suggestion search to select a customer. This unwanted and misleading message will not be shown any more.
626301	Unit templates only used default localization: The templates defined in Admin-Tool used for rendering units did not use the appropriate localization where it was available. Even if a template with correct localization for the client browser was defined, only the default localization template was being used. This error has been corrected and now the adequate localized template is used where available.
626839	Individual tickets could not be opened, if their history contained an e-mail entry with very complex HTML layout and high numbers of inline images. Such a history entry caused an error when rendering the entry. This problem has been resolved and such a ticket including the complex email entry will be rendered successfully.
626938	Unassigning engineer in ACF by keyboard not working: When trying to unassign an engineer in an ACF by pressing the backspace key this seemed to succeed and showed "not assigned" in the field after moving the focus, but the change was not saved to the ticket. When removing the engineer by mouse click, the change was saved. This incorrect behavior was fixed and both methods result in unassigning the engineer now.
627281	NIMH mail parsing exception: A deficient mail could cause an exception from NIMH while parsing it to a string. This problem was resolved and this exception does not appear anymore.
627346	A default value script failed in some cases when it tried to fill in the current engineer from the engineerService when creating a new ticket. This error has been corrected and the engineer can be set again.
627398	Dependent Enum scripts did not work with radio buttons: With newer versions dependent Enum scripts did not work in combination with visualizing the choice/data as radio buttons. This faulty behavior has been fixed which also includes that read-only radio-buttons will not trigger the script.
627497	Submitting an ACF after changing the main contact broken: It was not possible to submit an ACF and save the data after the main contact of the corresponding ticket had been changed before. The "OK" button did not react on clicking. This undesired behavior has been fixed so that the data modification can be saved even after changing the main contact.
627870	The indexer regularly stopped working after importing a scene which contained a deficient phone number configuration where the country prefix was null. This issue was resolved and such a problematic scene should not impair the indexer any more.
627955	Improper template management group ordering: The template groups in the web client template management could not be sorted in proper alphabetical order. Additionally the "All" grouping was not the first group list entry. This unwanted behavior was corrected and group ordering again behaves as expected.

627982	Possible exception when updating to latest version: When updating from a 6.9.3 version to the latest release of Consol*CM an exception could appear which prevented the system start after the update. This error has been corrected and the system reliably starts again after an update.

6.5 Known Issues

Number	Description
627286	Pasting (inline) images together with HTML text can still cause the images not displayed, if the image link cannot be resolved later on a different computer. In this special case images are not included but linked and depending on the client computers specific network accessibility the link address may in some cases not be accessible from this machine. In a future release improved handling of this kind of image pasting will be implemented.
628191	The error stack trace upon a failed LDAP login attempt is not handled even though a sufficient warning is written to the log files along with the stack trace.

7 Version 6.9.4.6 (15.12.2015)

Version 6.9.4.6 includes 6.9.3 versions up to 6.9.3.10, 6.9.2 versions up to 6.9.2.11, 6.8 versions up to 6.8.5.8 and 6.7 versions up to 6.7.13

7.1 Update and installation instructions

7.1.1 Security update

This release is a security update to provide countermeasures against potential deserialization attacks which can be put forward against the Apache Commons Collections library and others. Please see section 7.3.1 for additional information.

7.1.2 Data warehouse JMS communication channel not supported anymore

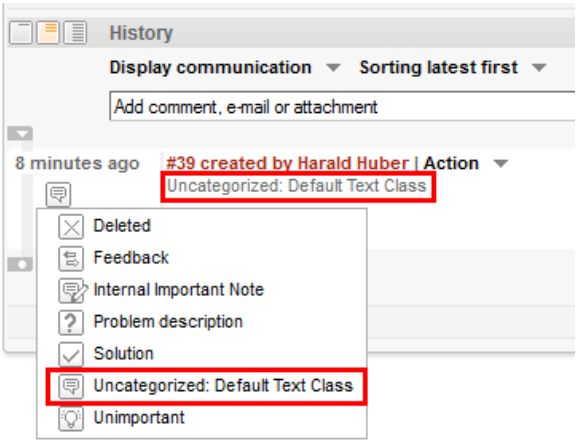
ConSol*CM does not support the JMS data warehouse communication channel anymore in version 6.9.4 or later, including this version 6.9.4.6 release. This has been stated explicitly for the JBoss 7 (JBoss EAP 6) application server platform when introducing support for this platform. However, it should be clarified that this also applies to the other supported platforms JBoss 5 and Oracle Weblogic 11g R1. When using a ConSol*CM version 6.9.4 or newer **ONLY** the DIRECT mode data warehouse communication channel is available. Please implement this change according to the available documentation when updating from earlier CM6 versions using the JMS communication.

No further instructions available.

7.2 New Features

7.2.1 Default text class available for manual selection (#622182)

Previously the default text class was not available for manual selection in the web client. This selection has been made available with this release. Now it is possible to assign the default text class to comments, incoming and outbound e-mails. The default value will show in the selector next to the entry just like other text classes, please compare the screenshot.



As previously, an existing text class can be designated as default text class by identifying it with the technical name as value of the system property *defaultContentEntryClassName* in the module *cmweb-server-adapter*. This new default text class will then be available as the default in the manual selection, too.

However, there is a slight variation: All the text entries which had no text class explicitly set will show the new default. The same is true for new entries. Those with a text class explicitly assigned will keep it, of course. So, if the previous default text class was explicitly assigned with the mechanism introduced here, the entry will keep this previous text class, too. If no class was manually set, then the new default class will be used.

7.2.2 Improved logging of data warehouse activities (#628400)

The logging of data warehouse activities has been extended so that there are several new log entries generated by this kind of activity. By default this additional logging is deactivated and will only show when setting the log level to TRACE in order to analyze data warehouse issues. These additional entries are written to the dedicated log file "dwh.log".

7.3 Changes

7.3.1 Security improvement: Deserialization attack countermeasures (#628576)

A very serious vulnerability has been recently (2nd half of 2015) discovered in the Apache Commons Collections library version 3.2.1 (please see for further details: https://issues.apache.org/jira/browse/COLLECTIONS-580) and others. The Commons Collections is a standard library which is used in numerous applications worldwide. It is also used in ConSol*CM where the communication between Admin-Tool and CM server as well as between the Process Designer and the CM server can be affected.

Objects of a specific library class might be used to build serializable collections which can execute arbitrary code when deserialized on the server side. Since this problem affects deserialization of data sent to the server it provides an attack vector in the earliest stage of processing.

A Java agent has been provided as immediate measure for all releases of ConSol*CM. This release and all subsequent ones now feature the adequate application code changes as product-immanent countermeasures to protect against this type of attack.

7.3.2 Changed privilege requirement for setting a default customer group (#628192)

When trying to set the default customer group on the engineer profile page, the customer group selection required the WRITE permission for this group. This is in fact unwanted since it may very well be reasonable to set a customer group as default without being allowed to create new customers for this group. This requirement has been modified and setting a default customer group now only requires the READ privilege for the desired group.

7.3.3 Notifications on receiving too large e-mails when using NIMH (#628361)

There had been no notifications or log entries when receiving e-mails that were larger than the maximum size configured while using NIMH for processing. This lack of information has been remedied and now there is an entry in the server log file on INFO level identifying the message which was too large. Additionally an error e-mail is sent to the administrator about the failed incoming e-mail processing.

7.3.4 DWH LIVE mode occasionally could not transfer tickets created by NIMH from incoming e-mails (#628250)

The data warehouse LIVE transfer mode occasionally could not transfer tickets which were created by NIMH from incoming e-mails. The LIVE mode transfer stopped operating when this problem occurred. It happened especially when ticket creation included scripts performing longer operations. This problem was already corrected in the higher CM version 6.10.1.0. The correction has been implemented in this release, too, so that the problem will not arise anymore.

7.4 Bugs fixed

Number	Description
627278	**First row of edit fields not aligned with the following ones** The first row of edit fields for general ticket data did not appear aligned with the next fields which belong to a different field group. They could begin left of the later ones for example. This has been fixed and in normal cases the edit fields are well-aligned now. There remain border cases, if the label of a field below is overly long without any whitespace. This will break the alignment, however adding whitespace to the label text could restore the alignment easily.
627479	**Log rotation for the indexer of JBoss EAP on Windows did not work** A log configuration problem on JBoss EAP with two different file handlers pointing to the same file caused the log rotation to break on Windows for the indexer. For this reason the indexer log file could grow indefinitely despite a log rotation configuration. This problem has been resolved and log rotation works since the two handlers point to two different files now.
627741	**Attachments with forward slashes in the file name were unavailable** Attachments of incoming e-mails which contained forward slashes "/" in their file name were not recognized/imported and thus were unavailable in the web client. This has been corrected and forward slashes "/" in the file name are now replaced by underscores "_" like in other e-mail clients. The attachments are therefore available in the web client now.
627882	**Resetting timezone information too late** Resetting or updating timezone information in the browser display occurred only after the second login, if the timezone configuration on the client had been changed. This problem was resolved and now a change in timezone information will reflect after the first login (except for some specific border cases when using the Firefox browser where you still need to login twice after resetting the system timezone for this change to work). So in order to apply the change in CM6 once the timezone setting on the operating system level has been changed a re-login is usually necessary and sufficient.
627981	**Wrong bullet list font size with non-standard default font size in editor** The font size of unordered lists was wrong when a default font size different from the factory standard was configured. The list font size still used the factory setting. This behavior has been changed and the list now uses the configured default font size as expected. The size of the bullet symbol may still follow the factory setting since this setting cannot be influenced.
628390	**E-mail addresses with a "%" character could not be entered** Entering e-mail addresses which contained a "%" character (per cent sign) was prohibited by the address validation even though this is a permissible character in an address. This was changed so that such an address can now be used for sending an e-mail without problems.
628430	**NullPointerException when calling operations on model objects from NIMH** When calling operations to change model objects like ticket custom fields a NullPointerException occurred in NIMH, if not an explicit model update call had been made in the mail processing script. This common problem has been addressed and now an automatic update call is made at the adequate stage of the processing chain.
628456	**Mails not fetched when NIMH configured to delete read messages** In case NIMH had been configured to delete read messages by setting the property "mailbox.default.task.delete.read.messages" to "true", mail messages were not fetched into CM6 at all. This error has been fixed and now the e-mail messages are fetched into CM6, then they are marked read and subsequently will be deleted with this setting.
628496	**Exception when changing the customer to a newly created one** Occasionally the engineer session was ended when trying to change the customer to one being newly created in this step, which was causing an exception. This error has been corrected and this procedure should always work now.
628529	**SQL exception when adding an additional engineer on Oracle** In the latest releases an SQL exception occurred when adding an additional engineer, if the database engine used was Oracle. This issue has been corrected and with this release it is possible again to add an additional engineer when using Oracle.

7.5 Known Issues

Number	Description
627286	Pasting (inline) images together with HTML text can still cause the images not displayed, if the image link cannot be resolved later on a different computer. In this special case images are not included but linked and depending on the client computers specific network accessibility the link address may in some cases not be accessible from this machine. In a future release improved handling of this kind of image pasting will be implemented.
628191	The error stack trace upon a failed LDAP login attempt is not handled even though a sufficient warning is written to the log files along with the stack trace (resolved in version 6.10.4.0).

8 Version 6.9.4.7 (21.12.2016)

Version 6.9.4.7 includes 6.9.3 versions up to 6.9.3.10, 6.9.2 versions up to 6.9.2.11, 6.8 versions up to 6.8.5.8 and 6.7 versions up to 6.7.13

8.1 Update and installation instructions

No further instructions available.

8.2 Changes

The keystore files have been updated. These files contain the code signing certificates for the Java applets for image pasting and CM.Office as well as for the Web Start applications Admin-Tool and Process Designer. Expired certificates cause an error message when starting the tools, depending on the Java version on the client machine. The new certificates expire in October 2019.

9 Version 6.9.4.8 (28.08.2017)

Version 6.9.4.8 includes 6.9.3 versions up to 6.9.3.10, 6.9.2 versions up to 6.9.2.11, 6.8 versions up to 6.8.5.8 and 6.7 versions up to 6.7.13

9.1 Update and installation instructions

No further instructions available.

9.2 Changes

9.2.1 REST API customer access to object restriction enforcement (#632053)

The REST API when used with valid customer credentials did allow accessing the data of other unrelated objects, if a REST request with a manipulated structure and a valid ID was crafted manually and issued. This access is undesired for most use cases. Usage of REST calls with engineer credentials will generally apply the desired access restrictions.

A new system property has been added which controls REST customer data access with customer credentials. It can be found in the module `cmas-restapi-core` and is called `security.restrict.unit.access.to.own.data`. It will be introduced automatically by the update with its value defaulting to “true”.

The new restricted unit object data access policy applied by the property value “true” will activate an additional check for requested customer data. The requested information will then only by returned, if either

- the requested item is the company for the customer logged in or
- the requested item is another contact of the company for the customer logged in.

Requests for other unit object data will get a response status `403 FORBIDDEN` in return. This policy is enforced for all requests for customer data, no matter if they are requested by ID or by search criteria.

Setting the property value to “false” will allow the less restricted data access as previously for backwards compatibility.