d.velop

d.3 admin: Administrator

Table of Contents

1. d.3 admin

1.1. Introduction

1.1.1. About d.3 admin

The application d.3 admin is used for the administration of all d.3 repositories in the network:

- Users
- Groups
- Advanced properties
- Datasets
- Document and dossier types
- Document directories
- Authorization profiles
- Document classes

Additionally, certain parameter of the d.3 server can be set to apply customer-specific configurations.

1.1.2. Hints on the user interface

Tooltips/quick-info: Placing the mouse-pointer over a button or icon and leaving it there for a moment will display a brief hint.

Context menus: Many elements on the surface are provided with context menus.

1.1.3. Prerequisites

This manual explains the administration of d.3 repositories and is exclusively targeted at d.3 administrators.

It is advisable to have read the manual of d.3 smart explorer and to have basic knowledge in Microsoft Windows to understand the content of this document.

1.1.4. Additional information

Additional information about d.3 admin can be found in the following documents:

Administration manual for dossier generation (d.3 admin folder scheme)

This manual gives attention to the dossier generation in d.3 admin.

Administration manual for d.3 async job inspector

d.3 async job inspector allows to monitor and control the d.3 async jobs being executed on the d.3 server in the background.

This extension of d.3 admin is a valuable help for the analysis of problems (e.g. in dossier link).

d.3 async job inspector expands the functionality of the existing joblist in d.3 smart explorer (**View | Job list** in d.3 smart explorer) with detailed comments about the d.3 async job. The display and editing of job parameters is another new feature.

You can use d.3 async job inspector to repeat and delete jobs or to change their priority.

The integration with d.3 smart explorer to display the job document and the connection to an optional d.ecs rendition service server (former d.tiff) fascilitate the problem analysis.

Administration manual for d.3 client config (client profiles)
The d.3 client config module is a plugin for d.3 admin, which is automatically configured with an installation of the d.3 applications (from version 5.5).

The client profile setup allows to individually customize and adjust the d.3 client (especially d.3 smart explorer and d.3 import).

For every d.3 istallation and every repository, individual profiles can be defined which can be assigned to certain user groups and which can then be selected on the respective client workstations. This allows you to configure default settings and limitations in the mentioned applications for the assigned user groups with a profile mechanism.

Thus, you design the user interface of d.3 smart explorer or d.3 import and adjust it to the users' needs.

Administration manual for the d.3 admin ldap
This manual deals with the LDAP module used to define d.3 users.

Administrator manual for d.ecs rendition service
This manual explains the d.ecs rendition service used to convert documents to TIF or PDF files on the server.

Administration manual for the d.3 search
This manual introduces the d.3 search module used for the full-text search in the d.3 repository.

Administration manual for d.3 abo service
This document contains information on the installation, configuration and the functionality of d.3 abo service.

Administration manual for d.3 explorer folder browser
This document contains information on the configuration of d.3 explorer folder browser.

1.2. Installing d.3 admin

1.2.1. Setup
d.3 admin can either be installed locally or on a network drive. The installation on a network drive is recommended to keep the effort low, while running an update.

> **Note**
> The distribution of d.3 admin with a d.3 application (like d.3 smart explorer) only works under certain conditions, as additional modules are located in the directory **AdminPlugIns** and sub-directories are not distributed with the d.3 application. If you distribute d.3 admin with the d.3 application, the plugins will not be distributed.

d.3 admin is installed on the server as part of d.3 core components. By default the installation is made into the directory **\d3\d.3 admin**. From there d.3 admin can be called directly. On sharing this directory you can access the file **d3admin.exe** located there.

If you want to install d.3 admin locally, this is possible via the separate setup of d.3 admin. An update must therefore always be executed parallelly on all workstations equipped with d.3 admin.

System requirements
Please refer to the central system requirements for d.velop products (on-premises). You can find deviating or more extensive system requirements in this documentation.

Warning
Support is only granted for systems matching these requirements.

Note
The d.3 admin can also be run on 64-bit systems.

Note
The operation of d.3 version 8 is only allowed in the combination server 8.x with client 8.x, where "x" is for the minor releases and these must also always be identical (example: server 8.1 only with client 8.1). A support for deviating system environment is not granted. Only during the update process a short-term, restricted operation (research, display (open) and storage) is allowed with a client of 7.2.2.

1.3. Starting d.3 admin

1.3.1. Start and login

Note
The login to d.3 admin is independent of the login to the d.3 application. Thus, you have to log in to d.3 admin, if you are already successfully logged into d.3 smart explorer (see manual for d.3 smart explorer).

The configuration file **d3login.ini** for the administration is located in the installation directory of the program **d3admin.exe** (**\d3\d.3 admin**). If the file is not there, you can manually copy it there:

- In case of an update installation you will find the file within the directory **\d3\d3client.prg\Admin_BAK**
- In case of an clean installation you will find the file within the directory **\d3\d3client.prg** (after a d.3 repository has been created)

1. Start the login dialog using the defined shortcut icon for d.3 admin (**Programs > d.velop > d.3 admin**).
2. Log in as an administrator of the d.3 repository.
3. Enter the following login information:
   - the predefined user name d3_admin
   - the respective password
4. Select the **d.3 repository** you want to login to in the dropdown selection.
5. With a click on the icon you get information on the connection data of the selected d.3 repository.

Warning
For the first login the password is also "d3_admin". It must be changed after the first login.

The password change can be accessed with a click on the button next to the password field.

The available login languages are **German** and **English**.

Click **Cancel**, to cancel the login process.

> **Note**
> With d.3 version 7.2 you can cancel the connection establishment to d.3 server, if you recognize that the server processes have not been started.

To confirm your entries and to complete the login process, press the **ENTER** key or click on the field **Log in**.

Alternative login with OpenID Connect

Using the pull-down button **Log in** you can choose between the standard login and the alternative login with OpenID Connect. For the login with OpenID Connect a configured and enabled identity provider is required. With enabled login via OpenID Connect you only have to select the repository and the language. The input fields for user name and password are disabled. Now click on **Log in**. Afterwards the dialog for the login with OpenID Connect is displayed.

The login variant is saved for all subsequent logins, so you don't have to select it again each time you log in.

If your entries were correct, d.3 admin will be started and the application window is displayed.

1.3.2. Logout

By quitting the application d.3 admin you will be logged out automatically.

To quit the application close the application window or press the keyboard shortcut **STRG+B**.

1.3.3. Update or clean installation - behavior in d.3 admin

When starting d.3 admin for the first time after an update or clean installation of the d.3 version 8, the d.3 system will be prepared for using the transport system. The following actions will be executed:

Clean installation

- Creation of the snapshot project

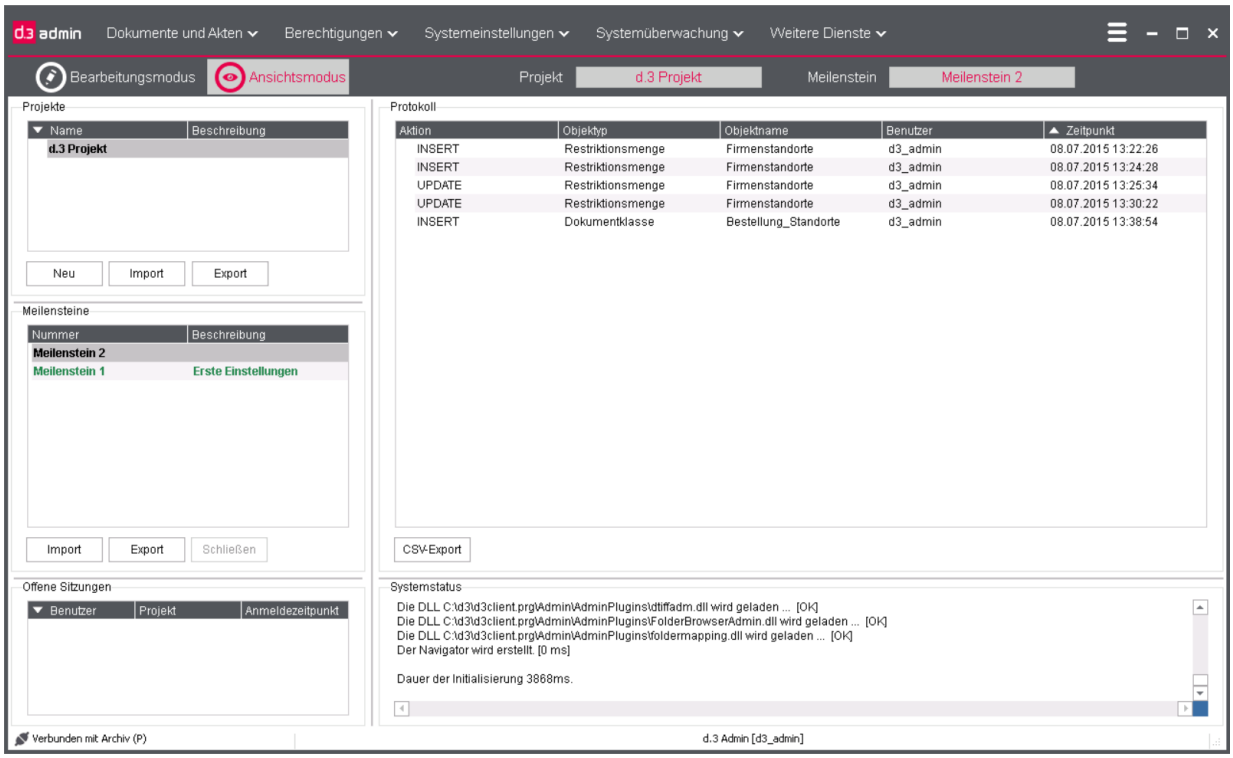
Update

- Generation of the synchronization IDs for the master data-objects
- Creation of the snapshot project
- Creating a first milestone in the snapshot project with the backup of the current master data
- Migration of the application profiles

After finishing these actions d.3 admin is in view mode. If you want to change anything, you have to create a project and switch into edit mode.

1.3.4. The main application window

The main application window of d.3 admin is divided into the following fields:

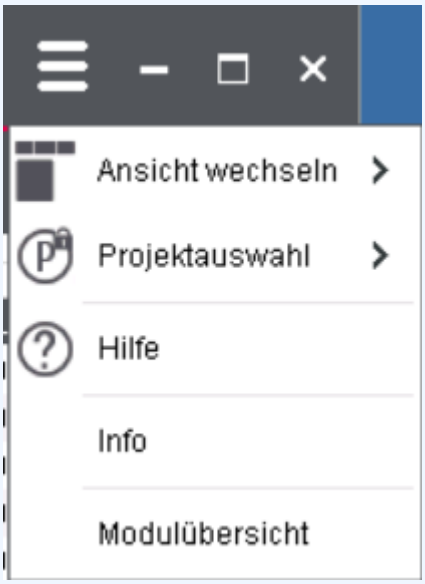


At the top right you can find the menu button as well as the buttons to minimize, maximize and to close the window.

Above the red horizontal line you see the Navigation and below the buttons for the Edit and view mode as well as the view for the current project and milestone.

Note

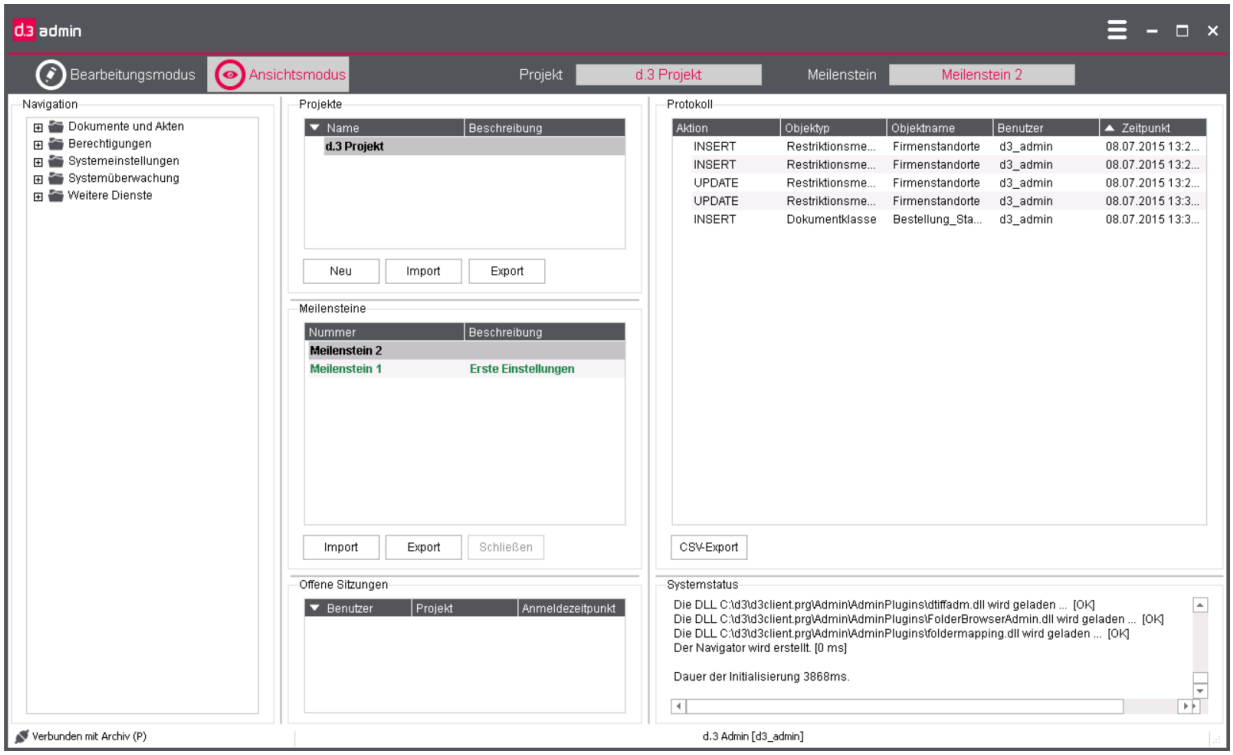
To switch to an alternative view, click on the entry **Change view** in the menu.



The fields **Navigation**, **Protocol** and **System status** can be displayed or hidden.

Projects, **Milestones** and **Open sessions** are always visible and cannot be disabled as these are urgently required to use d.3 admin.

For the other view a part of the navigation is displayed in the panel of the window below.



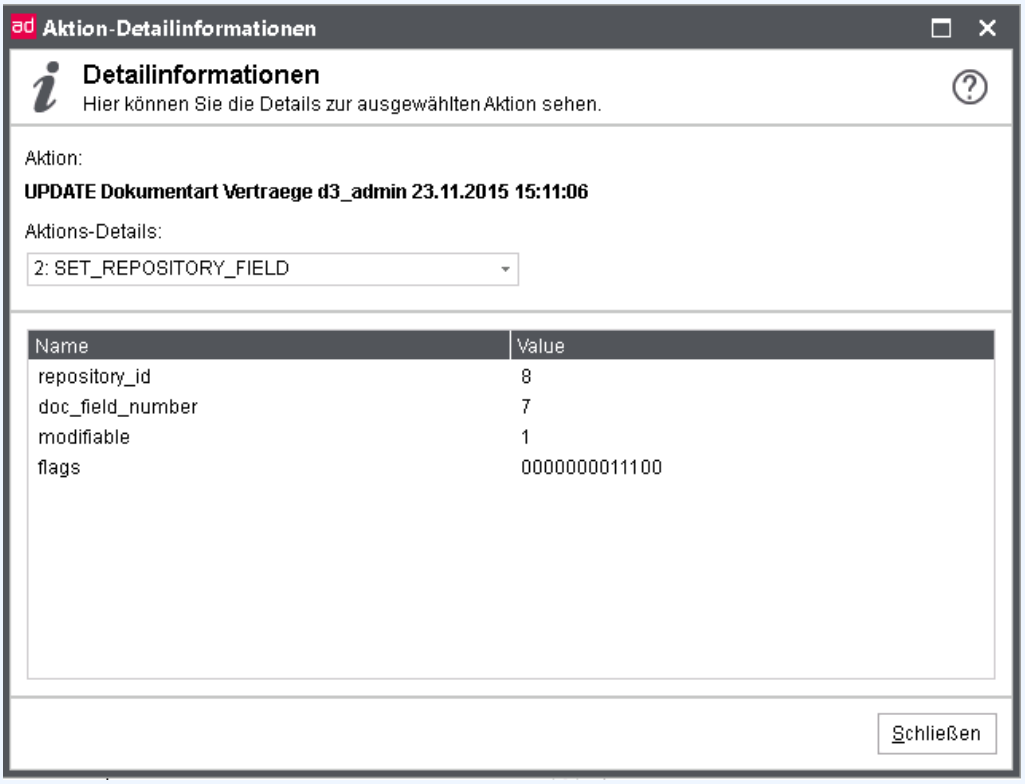
The application window provides also the following fields:

- Projects
- Milestones
- Open sessions (displays the currently logged in user including processing on which project)
- System status (shows current system messages)
- Protocol (lists changes currently made)

Note

The displayed changes of the currently selected milestone can be exported via the button **CSV-export** below the table as CSV-file.

Detailed information on the respective entry can be viewd via double-click:



1.3.5. Menu

Via the menu button you can find the entries **Change view**, **Project selection**, **Help**, **About**, **d.velop community store**, **Module overview**.

Note

You can learn more about current products and solutions and obtain them on the website of the d.velop ommunity store.

1.3.6. Navigation

The navigation consists of the main menu options **Documents and dossiers**, **Rights**, **System settings**, **System monitoring** and **More services**. Via the entry **Change view** in the menu you can decide, if you want to use a dropdown menu or the classic tree structure for the navigation.

Navigation - view as dropdown menu

Navigation - view changed

1.3.7. View mode/edit mode

The main application window of d.3 admin is opened in view mode by default. So the settings and the system status can be seen.

If you want to make changes, click on the button **Edit mode** in the application window.

Since you have not yet created a project when you first start the application after installation, you can also create a project in view mode.

To create a new project, click on the button **New** in the field **Projects** of the application window.

If you have the required administrative rights, you can make changes.

1.3.8. Projects

While working with d.3 admin, keep in mind that changes only can be done in the edit mode. To get into the edit mode, you need at least one project, which is in the status “open”. For each project only one administrator can switch into the edit mode. Several administrators can work in different projects at the same time, but never in the same project. “Open sessions” on the start page of d.3 admin shows you the projects, on which somebody is working on.

While switching from view mode to edit mode, you will be asked under which project you want to start your editing.

You can use projects to keep hold on certain configurations in d.3 admin. You can then export these projects and, for example, import them to other systems.

Create a new project and enter a name (the name has to be unique). Optionally a description can be entered. The project will be created in the status “opened” and cannot be closed by the administrator.

There are also closed projects, that are not displayed by default. Though it is either a created snapshot project automatically created by the system or projects created while importing transport files. These projects cannot be opened anymore.

See also: Notes on the transport system, Data backup

Create a new project

While switching from view mode to edit mode, you will be asked under which project you want to start your editing.

You can use projects to keep hold on certain configurations in d.3 admin. You can then export these projects and, for example, import them to other systems.

At the first start or to create a new project, Click on the button **New** in the field **Projects** of the application window.

Import a project

To import a project, you need a .d3trans or .d3backup file. You can look at some information on the milestones contained in the transport file before starting the import process. You do not have this option with backup files, as they are created independently of projects and milestones.

The import of a project or milestone file runs through the following steps:

- Implementation of a consistency check of the single files containing in the transport file. If it fails, an import is not possible. Changes on this files are not permitted.
- Performing a simulation of the import process to a transport file.
- To assure the synchronism of the data, it will be checked, if the milestone process to the project exists without gap in the target system. If not all milestones of a project are contained in the transport file, you have to make sure that the missing milestones were imported by a previous import.
- While writing the master data objects it will be checked, if a newer state exists, created by an import of transport files of other projects. In such a case the object will not be written again.
- The complete import process runs within one database transaction, so that any error (validation error or database error) leads to full rollback of the changes.
- If the project or milestones do not exist before the import, these will be created. The status of the project or milestone is “closed”.

This is how it works

1. Click on **Import**.
2. Select an already existing d.3 project file which you want to import into your repository.
3. Start the simulation of the import process with **OK**. After the simulation has been completed successfully, you will see an overview of all changing items and actions under **Transport data**.
4. Check all actions in the transport data overview.
5. Click **Start import** to perform the actual import.

> **Warning**
> Enable the option **Skip simulation** only if you are really sure what actions will be performed with the import of the selected file. If you enable the option, no simulation will take place before the import. The import will start immediately and you will not have a chance to check it beforehand.

Export project

You can export a project and one or multiple milestones.

> **Note**
> If you export the data for a project, all closed milestones will be bounded into the transport file. Select a storage directory and enter a description text for the export. This text will be displayed at the import again. As a suggestion the description in this dialog contains the project and the milestones to be exported. You can edit the suggested text.

This is how it works

1. Click on **Export** to export an already created d.3 project.
2. Select the directory where the file is to be stored.
3. Confirm the prompt with **Yes** to execute the export.

Afterwards you get the information that the export was done accordingly.

1.3.9. Milestones

With the milestones you can keep hold on certain changes in d.3 admin. You can then export these milestones and, for example, import them to other systems.

A milestone is always a part of a project and cannot be created independently by yourself. Also, a milestone cannot be deleted.

While creating a project, a milestone will be created automatically. Every milestone has a number, which begins at 1, for each project. Milestones are either in the status “opened” or “closed”. Only if a milestone

is in the status “opened” (it is marked as red in the milestone overview), changes are recorded in it. A milestone in the status “closed” (it is marked as green in the milestone overview), it cannot be opened again. For each project a maximum of one milestone in the status “opened” is possible. Though, this is always the last one created. There are also projects, in which none milestone is in the status “opened”. Changes cannot be recorded anymore in such projects. Such projects are created by the import of transport files from another d.3 repository.

The following data is recorded within a milestone:

- Datasets
- Document types
- Advanced properties
- Multiple languages
- Dossier generation
- Groups (except memberships of users)
- Document classes
- Authorization profiles
- Authorization profile assignments (not for users)
- d.3 organizational structure
- d.3 policy manager (not for users)
- d.3 restriction set manager (not for users)
- d.3 config (not for all parameters)
- d.ecs rendition service assistant

> **Warning**
> Files containing hook functions are not transportable!

You will be prompted to enter a descriptive text for this milestone. If you close a milestone (only possible in edit mode), the following actions will be executed:

- By means of the list of changes on master data objects all dependencies will be determined and the current state of these objects will be written into a transport file.
- This transport file will be imported into the d.3 repository (document type "Audit-Log").
- The current milestone will be closed and a new one is created in the status “opened”.

Close milestone

You can save or complete a milestone as follows.

This is how it works

1. Click on **Close**.
2. Enter a description of the milestone in the following dialog.
3. Confirm with **OK**.

> **Note**
> Enter a unique description for each milestone to be able to track which settings you have made in this milestone for a later import or export.

Import milestone

If you want to import an already existing milestone file into your repository, You can select and import this via **Import** in the following dialog window.

Read also Import project.

Export milestone

You want to export a created milestone.

This is how it works

1. Click on **Export**.
2. Select the directory where the file is to be stored.
3. Confirm the prompt with **Yes** to execute the export.

Afterwards you get the information that the export was done accordingly.

> **Note**
> If you export data for milestones, keep in mind that you can only select closed milestones. You have to select at least one milestone. This can also be several, whereby the list of milestones must be without a gap. However, when selecting milestones, you do not have to start at Milestone 1 or end at the last closed milestone. Again, select a storage directory and enter a description text.
>
> During the export process the data of the milestones from document type “Audit Log” will be loaded and merged to a ZIP-file. As file extension for the transport file .d3trans is used.

1.4. Administration

1.4.1. Plugins in d.3 admin

Various plugins of d.3 admin allow the configuration of the individual components.

The plugins can be found in the plugin folder of d.3 admin.

Many functions have to be provided (export from DLL) in order to load the plugins.

Moreover, the usage of the memory management ShareMem is mandatory, else undefined error messages can occur in d.3 admin later (protection faults, cancellation during release of the plugins).

1.4.2. Documents and dossiers

Datasets

A dataset is a group of values. This dataset can be assigned to an advanced property, which again could be linked to a document type. This would provide the user with this selection of values for this field in the document type.

This applies to the search as well as, more significantly, to the storage of new documents. The value list can be provided as a pure selection as well as a list of suggestions. This way the values of a dataset can also be used for a validation check. The according specification is made in the configuration of a document type.

Example: Let us suppose, you created a dataset with the name “Companies". This dataset contains the six entries

1. "Border Books"
2. "Leonard Krower & Sons"
3. "Simply Save"

4. "Total Sources"
5. "Opticomp"
6. "Sports Unlimited".

Afterwards you created an advanced property field named “Company". This field was assigned to the dataset “Companies".

Finally you create a new document type with the name “Maintenance”. This document type is to be used for the management of maintenance contracts in a software company. This document type is now assigned to the Field “Companies”. Thus, the document type “Maintenance” has only one property field. If a user now looks for a maintenance contract for the company “Opticomp", then he first selects the document type “Maintenance” from the document types. Afterwards, he selects the company “Opticomp" from the selection list (dataset) and starts the search for these maintenance contracts.

> **Warning**
> The name of a dataset may not contain any special characters with the exception of "+", "-" and "_". Other special characters like ":" are not permitted. Up to 250 characters for a value in a dataset are permitted.

In this example, the dataset “Companies" was assigned to the advanced properties field “Company". This dataset can, however, also be assigned to other advanced property fields. One and the same dataset thus be assigned to any number of user-defined advanced property fields.

Currently, four different types of datasets are supported. These are the manually created datasets, the datasets from a database and the datasets from a hook function and predefined datasets, e.g. the user/group list.

Manually created datasets

The type of datasets is defined manually. This dataset is always "static". If this dataset is used as a list of suggestions but the user specifies an alternative value, then this will NOT appear in the list of suggestions afterwards. The amendment therefore must be performed using the administration.

Examples for "static" datasets: Locations, departmental names, capturing sources of documents (e.g. file, e-mail, note, hint, protocol, ...).

To create a dataset, first enter the name for this dataset. This name must not exceed a maximum of 255 characters.

Assign a data type to this dataset. These are the same data types which are used when defining a field.

Enter the values to be assigned to this dataset.

> **Note**
> From version 7.1.1 you can determine, if the values shall be sorted or if the desired sort order shall be ascending or descending.

The values of the datasets can be translated in the additional languages to be supported.

Refer to chapter Multiple languages for more information.

With d.3 version 7 you can also use a list of numeric digits as values.

The actual specification as text is then applied in the translation of the dataset.

Example:

With the translation you can then define the actual texts.

Advanced properties

You can find under **Documents and dossiers > Advanced properties** an overview of already created advanced properties and their contents. This allows you to search for certain properties by the filter, a **Complete overview** and a **DB fields overview** can be viewed as well as advanced properties can be edited, copied or removed and new advanced properties can be created.

This overview of advanced properties is a kind of “clipboard”. All repeatedly required "fields" are kept here.

Let us suppose, your company has defined 10 document types, which use an a property named "Machine-Nr.". This property is linked to a dataset and undergoes a plausibility check. Without the “clipboard” function of the repository, you would have to define this field 10 times.

With the definition as an advanced property field, however, it is sufficient to configure the field once. Afterwards, this field can be used in an unlimited number of document types. If the dataset of this advanced property is changed later on, and the field exists in ten document types, then the dataset is automatically adopted in all ten instances.

An advanced property has various properties. These properties are described below.

Field properties

Title

The title of an advanced property is displayed to a user if he selects a document type. The title can have a maximum of 50 characters. Please define telling names. Whenever possible, choose familiar designations used in your company.

> **Warning**
> The name of the title must not contain any special characters except "+", "-" and "_".
> The special character “:” is not allowed. The d.3 repository in Unicode mode allows all available characters.

Data type

You must configure a **Data type** for every advanced property. Currently, the following data types are supported:

- **Alphanumeric**
- **Numeric**
- **Date**
- **Date and time**
- **Money**

Among other things, the data type affect the way, search queries have to be entered in d.3 smart explorer. Detailled information on this function can be found in the d.3 smart explorer manual.

Alphanumeric fields can contain any type of characters. The term “adopt” at this points refers to the entry of the property on import. This means, storage in the dialog (d.3 import) as well as the automatic processes (d.3 hostimport).

For the numeric fields, however, only the digits "0" to "9", the minus sign "-", the plus sign "+" and the decimal separator "." can be specified.

The fields of the data type **Money** can contain the same characters as the numeric fields.

The date fields are used to include a date.

Fields of the type date and time are usually normal date fields which, however, allow an additional specification of a time.

For the data type numeric and date you use the minus sign for the range-search or even for a search with an open range.

Warning

The properties must follow fixed syntactic rules using the batch procedures (**HOSTIMP, D3ASYNC**):

For numeric fields the decimal seperator “.” must be used.

The date fields must be provided in the German notation DD.MM.YYYY.

Note

The data type of an property can only be changed retroactively, if the property is not yet used in a document type.

Position

Before the preferred position of the advanced property can be explained, some general information on the storage of properties in the database must be given.

The d.3 repository stores the properties of a document in a database table which provides 89 columns for the storage of the property content. The data type of a field depends on the column it is stored in.

Database columns from – to	Field type
1 - 49	Alphanumeric
50 - 59	Date
70 - 79	Money
80 - 89	numeric

The number of the column is also used in a term called "**Database field**". The multi-value property fields are located in the columns 60 to 69. Thus, a field located in the column “33” is referenced to internally as “dok_dat_feld[33]”, a multi-value field “dok_dat_feld_60[x]”, with the “x” standing for the x-th item.

This storage structure of properties in the database limits the number of properties per document type or dossier type. One document type/dossier type can therefore contain a maximum of 49 alphanumeric fields, 10 date fields and so on. A document type/dossier type can never contain two fields (properties) at the same position, i.e. store the values in the same database-column.

Warning

A retroactive change of the DB position is no longer allowed for new repositories created with the d.3 version 7.

If the preferred position of an advanced property is modified, then the position of this advanced property field is only changed in this document type.

If you define a property field with a length of more than 35 characters but move this field to another database field in the document type which only supports 35 characters, the following thing happens: Since the advanced property definition specifies that the field is longer than the standard (35 characters), d.3 hostimport will encounter a database error. If the attributes are validated during the API import before, the respective field is truncated after 35 characters and the import succeeds. If datasets are affected, this can lead to validation problems. Thus, if it should happen that a field of more than 35 characters must be transferred to another database field, then this column must also be extended.

Length

The entry of a field length is only possible for alphanumeric fields. The default length for these fields is 35 bytes but can be extended to 250 bytes with a change of the database tables. An exception are the alphanumeric “multi-value property” fields. These fields are located at the preferred positions 60 to 69. The default length of these depends on the used data type. A length of maximum 250 bytes is predefined for the data type alphanumeric.

Warning

This is different for the fields 1-49: Since a direct change of the table **firmen_spezifisch** would occupy too much disk space, this table is initially still restricted to 35 bytes. Only if you want to create a field with a legth of more than 35 bytes, then only the respective column is extended.

To do so, simply enter the desired new length in the dialog, maximum 250. The database column is automatically updated to the desired length when creating the new property.

Note

Be careful with extending database columns. Meticulously plan it so that you do not occupy too much disk space for the data in your database.

The usage of different character encodings results in various disk spaces of single characters. Thus, you will read “bytes” in this chapter.

If the update is not executed automatically, then a database script **mod_doc_field.jpl_XX.jpl** (XX = repository ID) is created containing the necessary database-statements and which must be started via the d.3 server interface of the d.3 repository.

Do the following:

1. Sign in to the d.3 server interface.
2. Open the menu option **Programs > External Program** , choose the file **mod_doc_field_XX.jpl** and start it.

All necessary changes are combined here and are applied.

Multi-value property field

As mentioned before, multi-value property fields are a special type of advanced properties which can only be stored in the preferred positions “60” to “69”. For this reason these fields are also called “60-fields”.

The “normal” advanced property fields are used to store exactly one value such as an invoice number, the order date or the company name. A multi-value property field can contain various values.

Let us suppose, you need a field called “e-mail addresses” for a document type. You could configure several individual fields like “e-mail 1“, „-mail 2“, “e-mail 3” and so on. This implementation would however be very restricted. A multi-value property field can by default contain up to 100 values.

The d.3 configuration (**d.3 config**, section document storage) allows you to increase the number of entries in the 60-fields.

The maximum value for extension is 2000.

Since the maximum number of fields for a document type currently are 89 fields (properties), it would not have been possible to define the fields "e-mail 1" to "e-mail 100". A multi-value property field with up to 2000 fields would thus be the only solution.

> **Note**
> Multi-value property fields will be marked in the d.3 application as a table icon.
>
> You can also use the multi-value property fields in document classes.

Description

The description for an advanced property is for information only.

You can enter up to 255 characters. This descriptive text is also displayed in d.3 import, when the user right-clicks in the import form and the d.3 smart explorer and chooses the option **Properties**.

Existing data

You can assign a dataset to an advanced property field. The dataset can be provided by a static dataset, a database table (or database view) or a hook function.

Afterwards you can allocate a manually created dataset (see chaptermanually created datasets). Please note that only datasets of the same data type can be selected.

Database table

If you want to read values immediately from a database of the d.3 repository, specify the names of the table and column. Thereby, the dataset is marked unique. This way, you can only address one table in the d.3 repository. The data of the database table must either be entered by d.3 or via a customized function.

> **Note**
> However, you can also create so called database-views in your database system. Then, you enter the name of this view and the specified column name.

> **Note**
> In order to access data in a database table which is not part of the d.3 database system, you must develop a hook function (see chapter hook function).
>
> A maximum of 10000 entries are returned. If more are needed, a dynamic dataset hook based on the function **user_dataset_add_value** must be executed.

Hook-function

For this type of type of dataset you have to develop a hook function.

Hook-functions are customer-specific functions extending or changing the functionality of the d.3 server core. They can provide user-specific actions which are triggered by certain d.3 actions.

Hook-functions must be written in the interpreter language JPL (Jam Programming Language). This programming language is similar to the more widely know language “C” but significantly simpler to learn. However, it is not as powerful, either.

The document type “software order” is defined in your d.3 repository. This document type has the properties “Software company" and “Application". When archiving a document in d.3 import and searching a document in d.3 smart explorer the user should have a dropdown list for the property “Software company" from the values “Simply Save" and “Total Sources”. This can be easily solved by manually creating a dataset.

For the property “Application” the user should also be provided with a selection. In this case, however, the selection should be dynamically generated based on the property “Software company”. If the user selected “Simply Save” in the field “Software company” then the dataset should contain the values “File Storage Tool” and “Screenshot Creator”. For the value “Total Sources" the dataset should contain the values "Photo Editor" and "Total Sources” and “Video Editor".

This problem can only be solved by using a hook function which generates the dataset dynamically depending on the properties. Therefore, do the following in the d.3 repository:

1. Create an alphanumeric dataset named “Software companies" and the values “Simply Save" and “Total Sources".
2. Create an alphanumeric advanced property field named “Application” with the preferred position 1. Assign the dataset with the label "Software companies" to this field as existing data.
3. Create an alphanumeric advanced property field named “Application” with the preferred position 2. Assign the hook function with the label “BuildDataset” to this field as existing data.
4. Create a document type called “Software order". Assign to this document type the two fields created before.
5. Grant access to this document type to one or more user, e.g. in the form class <-> role or by adding them to a group.
6. Developing the hook function. How to develop and manage hook functions, you will learn in the d.3 Hook-training of d.velop AG.
7. Add the new hook function to the list of functions to be loaded by the d.3 server on application start. In order to do this, open the d.3 configuration and select the parameter **HOOK_JPL_FILES_CUSTOMER**. You find the d.3 configuration in d.3 admin below the **System settings**. Select the option **d.3 config**. Fill the parameter **HOOK_JPL_FILE_CUSTOMER** with the additional current value **hook_dataset.jpl**. If, for example, three functions were already defined there, select the fourth line by double-click and enter the value.
8. If you want to assort and load all values of the dataset by each access again, activate the checkbox in the picture of the hook datasets.

9. Afterwards save the d.3 config settings.
10. Close in d.3 admin.
11. Restart the d.3 server.

Source code

Below, you can find the source code of the hook function generated for the task described above.

This file can be generated in a simple text editor like Notepad.

The file has got the name **hook_dataset.jpl** and has to be in the application directory of the d.3 server.

The left column in the table contains the line numbers. These are not part of the source code. They are only included to describe the source code.

```
###
# Modul zum Ermitteln einer Wertemenge in Abhängigkeit von DOK_DAT_FELD_1
###
call d3_server_api_error_log("Hook-Demo: Modul geladen", 1, 9)
proc BuildDataset(repos_id, user, doc_type_short, row_no)
{
  call d3_server_api_error_log("Hook-Demo: Modul aufgerufen", 1, 9)
  call d3_server_api_error_log("Hook-Demo: DOK_DAT_FELD_1 =
:dok_dat_feld[1]", 1, 9)
  if (dok_dat_feld[1] = "Simply Save")
  {
    call d3_server_api_error_log("Hook-Demo: Simply Save", 1, 9)

    d3server_field_no_allowed  [1] = ""
    d3server_value_char_allowed[1] = "File Storage Tool"
    d3server_date_1_array      [1] = ""
    d3server_value_num_allowed [1] = ""
    d3server_repos_id_allowed  [1] = repos_id
    d3server_field_no_allowed  [2] = ""
    d3server_value_char_allowed[2] = "Screenshot Creator"
    d3server_date_1_array      [2] = ""
    d3server_value_num_allowed [2] = ""
    d3server_repos_id_allowed  [2] = repos_id
  }
  else if (dok_dat_feld[1] = "Total Sources")
  {
    call d3_server_api_error_log("Hook-Demo: Total Sources", 1, 9)
    d3server_field_no_allowed  [1] = ""
    d3server_value_char_allowed[1] = "Total Sources Photo Editor"
    d3server_date_1_array      [1] = ""
    d3server_value_num_allowed [1] = ""
    d3server_repos_id_allowed  [1] = repos_id
    d3server_field_no_allowed  [2] = ""
    d3server_value_char_allowed[2] = "Total Sources Video Editor"
    d3server_date_1_array      [2] = ""
    d3server_value_num_allowed [2] = ""
    d3server_repos_id_allowed  [2] = repos_id
  }
  return 0
}
```

> **Warning**
> Before developing your own hook functions you are urgently requested to attend a training at d.velop AG.

The lines 1-3 are comments and have no influence on the application behavior. Line 4 generates a log file entry in the logviewer. This row displays at program start of the d.3 server the entry “hook-demo: module loaded" in the logviewer. If you configured all settings correctly, then this entry should be displayed in the logviewer as soon as the d.3 server is started. In line 9, another logviewer entry is generated. The content of **doc_field_1** will be displayed here. It is important to use the “:” before the name of the field. Otherwise the entry would display the text “doc_feld[1]” instead of the field content.

Now, in row 11 the content of **doc_field_1** will be requested. In case of the term “Simply Save”, the rows 14 - 26 will be run through and in case of the term “Total Sources”, it will be the rows 30 - 44.

The command in line 15 has to be given for compatibility reasons. In row 16 the first entry of the dataset will be assigned. Because in our example we want to assign the dataset to a alphanumeric advanced properties, it is necessary to use the global list (array) **d3server_value_char_allowed**. In case of a date field, the array must be **d3server_date_1_array**, while it would be **d3server_value_num_allowed** for a numeric value. The date must be entered in the German notation “DD.MM.YYYY”.

Finally in row 19 the ID of the advanced property field will be assigned to a global array. This ID is passed on to the hook-function by the d.3 server during the function call (line 6).

Further literature on the topic of “hook-development" can be requested from d.velop AG.

Predefined dynamic dataset

The additionally available dynamic datasets include the users, user groups as well as already entered values (dynamic datasets) as selection options.

All users: You receive a list of all at runtime existing users of the d.3 repository.

All groups: Equivalently, all groups of the d.3 repository are listed by this assignment from the existing data.

All users/groups: All users and groups of the d.3 repository will be provided for selection as dataset.

All existing values: The database table is read and every existing value of the property field (of the selected document type) is shown.

This functionality had to be partially implemented using hook functions in previous versions.

> **Note**
> From the database table, all values of the referenced document type are read, reduced to distinct occurrence and then displayed in alphabetical order.

Plausibility check advanced properties

The plausibility check for an advanced property field serves to test the validity of property values entered during the import process.

Let us suppose, a user wants to archive a document of the type “Invoice”. The document type contains an advanced propertiy called “Company” in which the user has to enter the company name from the respective invoice. At this place it is to prevent that the user enters names like “12345” as a company name. As a company name at first character a ["A" - "Z"] followed by ["a" - "ü] n times is admissible.

Without a plausibility check, the invoice would be stored under a wrong company name in the d.3 repository. A targeted retrieval of this document would therefore be almost impossible.

In the application d.3 import this plausibility check is automatically executed when importing. However, to optionally start the validation manually, select the function **Validate input** via the context menu, the link in the navigation panel or the button.

> **Warning**
> The plausibility check works in the automatic import processes like hostimp as well in the manual import procedures using d.3 import.
>
> However, if the configuration parameter **NO_ATTRIBUTE_CHECK_ON_BATCH** is set to "**Yes**", then no plausibility check is performed for the automatic import procedures. This parameter can be set in d.3 config.

Hook-function

As mentioned in the chapterDatasets, a hook function is a routine written in JPL, which is executed by the d.3 server. In the context of the plausibility check, hook functions can also be used. Within the hook function itself you can execute complex validation procedures. For example, you could check whether an entered value (such as the company name) exists in a specified database table Plausibility check).

Enter the name of the hook function in the settings of the advanced properties. The further procedure is similar to the one described in the chapter on datasets. As described there, you must create a text file with the names of the defined hook functions. Several hook functions can be entered in one file. Hence, you could also enter the program code of this function in the file **hook_dataset.jpl** (see chapter Assigning datasets from a d.3 hook function). In addition to that, the file **hook _dataset.jpl** must be added to the list of loadable hook functions if not already done).

Minimum/Maximum

The check for a **Minimum** and **Maximum** only refers to numeric advanced property fields.

If the input of a numeric field has to be between 500 and 1000, you would enter 500 as the minimum value and 1000 as the maximum value.

Regular expression

Regular expressions are widely used in the Unix-environment to search and edit text. They serve to describe text patterns in a formal language.

In d.3, regular expressions are used to limit the entry of text in a property during the import to certain patterns. The following sections provide basic information on "single character patterns" and "multipliers" and are illustrated with examples.

> **Note**
> Regular expressions only apply to alphanumeric fields in d.3. They cannot be used for numeric fields, date fields or currency (money) fields.
>
> Further information about regular expressions can be found in UNIX reference literature or on the Internet. However, some expressions of the UNIX-specific dialects like anchors and braces will not work in d.3. Always check the new syntax with the test feature.
>
> You can define the interpreter of the regular expressions (POSIX, PERL, JPL) in d.3 config. Thereby the allowed syntax can be significantly different.

Single-character-patterns

The basic element of the regular expressions is the single-character-pattern. When you use a single-character-patterns as a regular expression in d.3 then no other character than the one specified in the single-character-pattern itself can be entered.

Regular Expression	User input	
A	A	
A	B	
A	Easy	

Amount

Several characters or ranges can be combined to a set. The sets are entered in square brackets. A set in its entirety also serves as a single-character-pattern. This means, when using a set as a regular expression, the input is valid, if one of the characters in the set was entered.

Regular Expression	User input	
[abc]	b (lower-case)	
[abc]	D	
[0-9]	F	
[0-9]	7	

Negation

Vice versa you can enter a set of characters as set , that is not allowed in the property value. To do so, enter as first character “^” in the value bracket. Please note that regular expressions differentiate between upper- and lower-case characters!

Regular Expression	User input	
[^abc]	D	
[^abc]	a (lower-case)	

Patterns

Finally, you can combine several single-character-patterns into a pattern. These patterns can also include sets. Ranges are validated against their ASCII value. Special characters like the German “Umlauts” or French accents have to be included in the pattern or they will not be accepted.

Regular Expression	User input	
[A-ZÄÖÜ][a-zäöü][a-zäöü][a-zäöü]	John	
[A-ZÄÖÜ][a-zäöü][a-zäöü][a-zäöü]	Saw	

Regular Expression	User input	
[A-ZÄÖÜ][a-zäöü][a-zäöü][a-zäöü]	Administrator	 (only exact four characters are allowed)
[A-Z][a-z][a-z][a-z]	Saw	 (ä is not allowed)

Colon (.)

The colon (.) is a special regular expression. It serves as a wildcard for any character except #013 (carriage return).

Regular Expression	User input	
[A-ZÄÖÜ].[0-9].	BE13	
[A-ZÄÖÜ].[0-9].	A999	
[A-ZÄÖÜ].[0-9].	A1	

Multipliers

Multiplier characters designate how often a single-character-pattern can be entered:

Any number, including 0 times:	If a character should be permitted to be entered between 0 (not at all) and ? (as often as desired) times, you can add the * (Asterisk) to any single-character-pattern ("*")
Exactly n time(s)	if a exact number (n) of characters should be entered append the characters \{n\} to the single-character-pattern and substitute the number for the n.
at least n time(s)	if a character should be entered at least n times, append the characters \{n,\} to the single-character-pattern and substitute the number for the n.
Between n and m times.	if a character should be entered between n and m times, append the characters \\{n,m\\} to the single-character-pattern and substitute the minimum number for the n and the maximum number for the m.

Note

If for interpretation of the regular expressions the "POSIX Extended Regular Expression Syntax” is used, leave the backslashes “\”.

Regular Expression	User input	
Alpha.*	Alpha.*	
Alpha.*	Alphabet	
Alpha.*c	Alphanumeric	
x\{3\}	X	
x\{3\}	xxx	

Regular Expression	User input	
Yes\4\}!	Yes!	
Yes\4\}!	Yeeees!	
Cancel\{2,3\}ation	Cancelation	
Cancel\{2,3\}ation	Neighbour	

The button **Test** can be used to check a sample entry against the regular expressions. Enter a string in the field and click on **Test**.

> **Note**
> The validation of the regular expressions is executed in a JPL interpreter. This interpreter truncates all appended spaces from a value in a property field. That means, trailing spaces in a property field are deleted before the validation!

Complete overview of all advanced properties

Use this function to display all advanced properties defined in the database together with their property fields in one table. To display the overview table, do the following:

1. Open the window **Overview of advanced properties**.
2. Right-click on any field.
3. Select the context menu entry **Complete overview**.

The assignments show which advanced properties were mapped to which database field. Duplicate mappings can also be recognized.

Overview of database fields

Use this function to display all advanced properties defined in the database together with their property fields in one table. To display the overview table, do the following:

1. Open the window **Overview of fields**.
2. Right-click on any field.
3. Select the context menu entry **Complete overview**.

The assignments show which advanced properties were mapped to which database field. Duplicate mappings can also be recognized.

Document types

The central characteristic for classification of documents in d.3 environment is the document type. All documents (e.g. a Microsoft Word document, an AutoCad drawing, a sound file, an executable file or a video clip in d.3) are assigned to a document type or dossier type when stored or created via the d.3 import or the Hostimport). For the end user, the document type is the most important criterion for searching for documents.

Note
You must make sure, that you create document / dossier types, which match to your company. Use names which are familiar in your company.

Warning
You can define up to 2,000 document-/dossier types for the d.3 repository.

Note
The overview of document types does not display the system document types. However, these are still available for the definition of document classes.

Properties of document type

For the creation of a document- or dossier type, the following information must be entered. You are guided through the process in various dialogs.

Basic configuration

You have to allocate further information besides the name for each document or dossier type.

Name

For the user, the name of a document type is the most important property to differentiate between the document types. You will find this name in all d.3 applications.

Note
The name or long name of a document type can have a maximum of 50 characters.

Warning
The name of a document type may contain all characters. The names of the document types must always be unique.

Note
Changing the name of the document-/dossier type afterwards requires a user to re-login to the d.3 repository.

The user is signaled to do so with a blue exclamation mark overlayed over the d.3 icon in the notification area (systray). Alternatively the symbol is displayed in the result list.

d.3 smart explorer can not create the display in this case, because the structure of the new document type is not loaded.

Warning

Please note that the change of existing names can lead to problems. Problems always arise when processes use the name of a document/ dossier type in your d.3 environment for internal procedures. This is the case when using the product d.cold of d.velop AG.

The same applies for the module d.3 hostimp, which might be configured to use the name of the document or dossier type. The name can also be used in hook functions.

You should therefore always try to use the 5-digit short name of a document type for reference. In contrast to the document name, the short name will remain constant for the entire lifetime of a document type. The name of a document type, however, can easily be changed and translated.

Type

The d.3 environment differentiates between document types and folder types.

The most significant difference is that dossiers can contain subordinate and superordinate documents.

In dossiers, usually no document is directly stored as they are only intended for the structured storage (as accustomed from paper archives).

Short name

The short name of a document-/dossier type is the unique identifier and the most important characteristic for processes and actions in the d.3 environment. The short name differentiates between lower- and upper-case characters. Hence, the two short names XFILE and Xfile designate two different types. The maximum length is five characters. The short name may only contain the following characters: "a..z", "A..Z", "0..9" and the minus character “-”.

Proposal for using a short name:

- Use all five digits.
- Use upper-case only.
- Select “F” for a dossier type as first or last character, “D” for a document type.

As the short name is displayed in case of errors in the central protocol, this significantly helps you at troubleshooting.

Note

The short name of a document/dossier type cannot be changed later. You must delete the document/dossier type and create a new one under the new name. In order to be able to remove a document/dossier type you must first delete the documents assigned to this type.

Warning

A change of the short name (Remove + Add) can lead to problems. Please note that processes in the d.3 environment might be using the short name for internal procedures. Among these processes are d.cold, the d.3 hostimp or hook functions.

Repository ID

The d.3 repository supports several secondary storage systems via d.ecs storage manager which is controlling them. The secondary storage is addressed via d.ecs storage manager. You can store the documents in individual "logical" archives on the secondary storage medium.

For example, the documents of the type "Deliverynote" are to be stored in one logical archive (such as "DA" for "Deliverynote Archive") while those of the document type "Invoice" are stored in another one (such as "IA" for "Invoice Archive"). For this effect, the repository ID is used.

You can enter a maximum of two characters for the repository ID. Upper- and lower-case characters are supported. Currently, the number of repository ID's is limited to five. The allowed IDs will be adjusted at the d.3 config in the section document storage. There, select the parameter **server_archive_id_array**. These settings can also be accessed from the point **d.3 Document Storage** choosing the option **Additional Server IDs**.

The unique identifier of documents (document ID) is generated using the repository ID. If, for example, you choose the repository ID "IN", then a document ID could be "IN500000".

Even if you are not using secondary storage, you can assign an repository ID. This way you could classify the document types and prepare the system for a later use of a secondary storage system. Once installed, the documents could then be moved to the secondary storage using this existing classification.

> **Warning**
> If you are using different repository ID's you must configure d.ecs storage manager as well. A detailed description of the possible settings can be found in the manual on d.ecs storage manager. An adjustment is also necessary if you subsequently change a repository ID, otherwise the documents are not stored.

Unprotected web access

This setting is needed, if you want to use d.3 webpublisher by the d.velop AG .

This product enables the access to documents in the d.3 repository via the World Wide Web. The user does not have to authorize against the system here. There is therefore no query of the user name or password.

The software d.3 web publisher is not supplied with version 6.3 since it used the outdated technology of CGI scripts.

On opening the URL for d.3 webpublisher, all documents/dossier types which have been configured with the option **Web published** are displayed for search.

Within the web published document/dossier type, however, only those documents are displayed, which were previoulsy marked as published in d.3 smart explorer.

In order to publish documents in d.3 smart explorer, the file **WebPub.DXP** must be located in the working directory. If this file does not exist, you can either include it in the d.3 application distribution or manually copy it to the working directory. After copying the file manually, you must set its file property to "hidden" (command line call: **attrib +h WebPub.DXP**), to prevent a deletion with the next start of d.3 smart explorer.

Once the file exists in the directory you must restart d.3 smart explorer. Then, you can retrieve the documents to be web published in d.3 repository and choose **Web published** from the context-menu of the document.

Keep in mind that publishing is only possible for released documents.

If the menu entry **Web published** does not exist in d.3 smart explorer, then either d.3 smart explorer was not re-started or the file WebPub.DXP is not located in the d.3 smart explorer working directory.

(4-eye-principle)

If this option is active for a document or dossier type, the owner of a document assigned to this type is not permitted to verify a document himself (status Verification -> complete verification) even if he

has the necessary rights. The editor is always the one holding the document in his processing while the owner is the user who stored the document in d.3 who created the release version in case of multiple versions.

This means, the editor (creator/modifier) of a document must be another person than the one verifying it.

Advance booking

The booking of documents is a special procedure of d.velop AG.

This procedure is used to create properties for a document, before the actual document is archived. In this procedure, the document is archived some time later (e.g. **late capturing** with barcode).

This procedure is used by d.3 import. If you change into the booking mode in d.3 import, only those document or folder types are displayed for selection, for which the option **Book** was set. Additionally, no document is stored whit a click on **Book** but only the attribute data are created in the d.3 repository.

Furthermore, a barcode cover page can be printed, which will be laid over the “analog” document at the clerk and afterwards it will be sent to the central scanning department. Using the barcode, the document is then mapped to the existing properties.

Please refer to the customizing manual for d.3 import to find out how the booking mode can be enabled.

During booking, an entry is made in the database table **firmen_spezifisch** when the properties (characteristic data) are created. By default, the entry in the database **dok_dat_feld_88** is made. This default setting can be changed using the d.3 configuration. To do so, choose **System settings > d.3 config >** Import Settings > **field number for barcode status** the parameter **BARCODE_STATUS_FIELD_NR**.

When creating the attribute data, the value "0" (zero) is entered in this database column. This value specifies that no document exists for these property data yet. When the document (file) is imported at a later time, this entry is changed to the value "1" (one).

Time to live

Since there may be a different retention requirement for each type of document, you assign the data for retention here. The default setting is for the retention period is 50 years.

For the GDPR-compliant planning or configuration of your d.3ecm system, however, it may be necessary to change this setting. You should therefore inform before creating document types with the requirements of GDPR so that you can set the correct retention periods. An ECM system like d.3ecm can only be set up or put into operation when the retention periods for the individual document types have been clearly defined and configured. A subsequent change of the retention periods on the secondary storage - after initial configuration and operation - is extremely costly or even impossible.

Please also check whether the desired settings for deleting documents/objects are set in d.3 admin under d.3 config. Please note the following settings:

- „Logical“ Deletion of Expired Documents
- „Physical“ Deletion of Expired Documents
- Delete Documents from Secondary Storage

> **Note**
>
> If a retention period other than eternal is specified for a file, the warning message appears: "To avoid unintentional, automatic removal of dossiers, the retention period of a dossier should always be “forever”."
>
> Please note that the retention period can not be "0".

Time to Live in Month

The lifetime is added as a value to the current date with every new version of a document and written as the retention period in the database.

You can define for each document type, if you want to use the **Eventless deletion** (previous default setting) or the **Event-based deletion**.

Lifetime from

You can now define for event-independent deletion when the lifetime should begin.

By default, the lifetime will be specified from the import date, that means the count of months will be added to the import date and sets the retention period.

The lifetime can also be projected by the end of the selected month to comply with special requirements, e.g. the end of the quarter oder fiscal year.

Write to secondary storage

This option specifies that the document/dossier types are to be written to the secondary storage system to store them in an audit-proof manner. Only the documents in the d.3 status "Release" and "Archive" are written to the secondary storage.

> **Note**
> This setting does not effect the automatic storage of documents by itself. Additionally, you need to set the parameter **AUTO_DOCS_ON_SEC_STORAGE** to "Yes" in the d.3 configuration. This setting can be found under the section **Secondary storage**. Please also check the other settings for addressing the secondary memory. Furthermore you have to have installed and configured the product d.ecs storage manager. A jukebox with a jukebox controller is supported.

> **Warning**
> If you enable the secondary storage for a document type and also apply the settings in the d.3 configuration but no secondary storage exists, then a copy of the document is written to the cache directories for the status "**Release**" and "**Archive**", but is not deleted automatically due to the "missing" receipt.

> **Note**
> The components d.3 async and d.ecs storage manager must work correctly for the out-sourcing of documents to run smoothly.

If the option **Write to secondary storage** is configured for a document type and you perform a status transfer (e.g. in d.3 smart explorer with a right-click and selection of **Status transfer > Release**, then the document is persistently written to the secondary storage.

Remove by event

If the event-based deletion is selected, then the retention period is determined via a defined event such as a product cancellation.

For more information see chapter Retention periods and deletion.

The actual deletion requires additional parameters in the d.3 configuration to be set (see d.3 Asynchronous processing).

Lifetime in cache

If a document is transferred into the status "Release" or the status "Archive", either directly by creating the document with d.3 hostimp into the status "Release" or manually in d.3 smart explorer, then the document is copied to the **Jukebox** exchange directory on the d.3 file server, as the settings for the option Write to secondary storage were applied.

If the document was copied suscecssfully to the secondary storage system, then the document remains on the file server for the specified time.

If a deleted document is requested by a user, it is transferred from the secondary storage to the d.3 file server again. On the file server this document can be found in the exchange directory **Jukebox**. There, the document will be provided until d.3 gateway deletes it again. The detailed settings for the deletion of the documents in the **Jukebox** directory can be found in the manuals for d.3 gateway and d.ecs storage manager by d.velop AG.

The retention period in cache in days does not affect this deletion in the **Jukebox** directory.

> **Note**
> The components d.3 async and d.ecs storage manager must work correctly for the out-sourcing of documents to run smoothly.
>
> Furthermore, a correctly configured d.3 gateway is needed for the retrieval.

Early removal

> **Note**
> If the free hard disk capacity on the d.3 file server becomes too low, documents already written to the secondary storage system ("burned") will be deleted from the hard disk in any case, regardless of the setting **Cache retention time**, if this was done with the option **Allow removal in case of lack of storage space**.

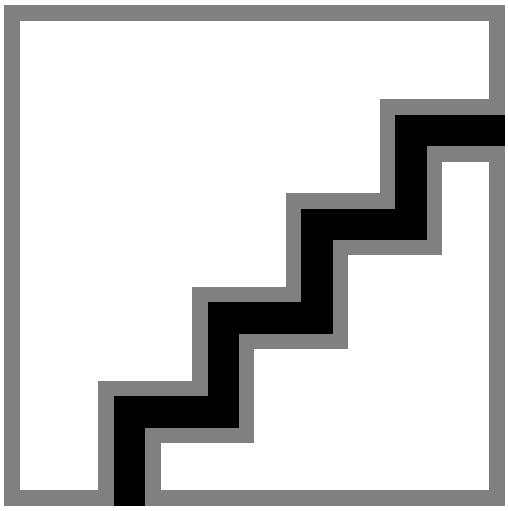
Cache retention time: forever

> **Note**
> From d.3 version 6.3 you can determine to prevent documents from being deleted from the cache at all (cache retention time: forever). However, this entry has to be used with care. It should only be applied to document types with low data load. Therefore, you also receive a warning message.
>
> The cache retention time cannot be "0" anymore.

Definition for color marking

You can define a color marking for the document type by clicking on **Add** or by performing a double-click in the clear zone.

A window “Color marking” opens, which displays the color palette and provides a text field for a description and language selection.



1. Select the desired color from the color palette.
2. Optionally, enter a descriptive text.
3. Change the language, if needed.
4. Confirm your color marking with **OK**.

> **Note**
> Color markings are some type of customizable feature for flagging documents as each company can internally define the meaning of the colors. Similar to real-world dossiers (populated with pages made of paper) you can now add a kind of sticker in a specific color to a document.
>
> There are 24 colors available. It is recommended, e.g. to define a number of colors as signals for specific company-wide or departmental processes and to leave a number of colors for individual purposes for d.3 users. For example, a certain color means that a document needs revision, needs verification before printing, or is part of a specific workflow step.
>
> Furthermore you as a d.3 administrator can also define for each category (document type or dossier type) each color a text, that is displayed to the users, if they show the mouse pointer on it.
>
> The text will only be displayed, if you assign the color to a document or a dossier in a certain category. The same color can also have another text in another category.

Selecting advanced properties

Every document- or dossier type is assigned to between 1 and 89 advanced properties. These assigned advanced properties are filled with the properties for the documents of this document or dossier type during the d.3 import. If you later add documents to the d.3 repository, either in dialog or in batch mode, these properties are filled with values. When searching for documents, you can explicitly search for these values.

You can allocate maximum the following properties to each document type:

- 49 alpha-numeric properties
- 10 date properties
- 10 money properties
- 10 numeric properties
- 10 multi-value properties

> **Note**
> If you subsequently change the assignment of the extended properties, it is necessary for users to log on to the system again.
>
> The retroactive change of this assignment does not result in a loss of this information, even if a large number of documents have already been imported with a different mapping. If you restore the previous assignment, all previously entered properties are displayed again in d.3 smart explorer.

The dialog **advanced properties** allows to assign the advanced properties to the document type.

1. To do so choose property von der rechten Seite aus **Available advanced properties**.
2. Add to the document type on the left side **(< Add, << Add All)**.

If you want to remove an advanced property, proceed as follows:

1. Select it on the left under **Selected advanced properties**.
2. Click on **Individually >** or **All >>**.

Position of advanced properties

In order to influence the displayed order of columns in the result list in d.3 smart explorer or of the fields in the property dialog for a search or import, proceed as follows:

1. Mark a field entry in the list.
2. Move this entry as you want with a click on the buttons.
3. You can simply change the position by drag & drop, "picking" a row with der left mouse button and drawing it to the corresponding position.

> **Note**
> Please note that the manual repositioning of columns by the user in d.3 smart explorer might override these defaults. In d.3 smart explorer however it is also allowed to restore the default positions under **View | Reset column arrangement** .

PropertiesThe following context menu options are available when selecting the property:

- Usage
- Modifiable Yes/No
- Hideable property

> **Note**
> There are no problems if you change the display positions subsequently. However, it is necessary for users to log on to the d.3 applicationanmelden.

> **Warning**
> A retroactive change of the database position is **no longer** supported by d.3 version 7 in a new created d.3 repository.

Manage properties

To get an overview of the properties in the dialogs **Overview of document types**, **Advanced properties** and **Positions of advanced properties** , open the context menu and select the option **Properties**.

In this window you can find information on the selected property.

Click on the buttons **< Back** or **Next >,** to navigate to the other properties of this list.

Usage of the property field

To get an overview of the document types and classes which use the property field, open the context menu and select the option **Usage**.

In this window you can find information on the usage of the selected property field.

Click on the buttons **< Back** or **Next >,** to navigate to the other properties of this list.

Give details for the property

To change the property, mark an entry in the positions of the advanced properties and click on the button **Properties**.

Alternatively: Open the dialog **Properties** by double-clicking.

This opens the editing dialog.

Database position

From the d.3 version 7 a retroactive change of the database position is not possible. For an update to version 7 it is considered, that this was possible in older versions.

> **Warning**
> The following explanatory notes ONLY apply to d.3 repositories, that were created by an version older than 7.

Alphanumeric fields 1 to 49This option changes the database position of the property field.

When changing the database position you must take account of the requirements for property:

- Date fields 50 to 59
- Multi value-fields 60 to 69
- money fields 70 to 79
- numeric fields from 80 to 89.

> **Warning**
> A retroactive change of the database position can be forbidden via a server parameter!
> Thus, all positions should be worked out and defined meticulously in the planning phase!

Mandatory

If the parameter **Mandatory** is set on “Yes”, then the field has to be filled by import (manually oder hostimport) of the document. If the field is still empty at the storage of documents, an error message will be produced.

> **Warning**
> Hidden fields should never be made mandatory at the same time since the import of such a document might then fail!

Modifiable

If this option is set on “Yes” (standard), then the property value can be edited after the import of the document. Else the change will not be applied.

If the property is thus defined for a document type as "protected" property, then its value can only be edited by an eligible user.

Title fields

Title fields are used to improve the display of document properties in d.3 smart explorer.

If you open a document in d.3 smart explorer and select the context menu option **Display Links**, then you would usually see the document ID together with the name of the document type, e.g. "T000065565 Supplier Invoice".

If you want to use more descriptive identifiers, then you can choose one or multiple title fields. d.3 version 7.2 allows you to define different d.3 system fields as title fields.

Hideable

This function allows you to prepare the properties in a document type for later hiding.

Via the rights management of document classes or profiles you can show the hidden property again.

This function is also available in the context menu > **Hideable property** or via the **Properties** dialog.

If a property is marked as hideable, then the icon in the overview is transparent:

In the overview of document types you can see that one property is marked as hideable:

As you can see in the illustration, the column **Hideable** must feature a "Yes" for a property to be hidden later, else the column contains "No".

> **Note**
> By default, the value **Hideable** is set to "No" for all properties.

> **Warning**
> If you hide a mandatory field, then the users cannot fill this field with "content" on import and the user will receive an error message!
>
> Never hide properties which are also marked as mandatory fields!

To display hidden properties for individual users again, you must grant the right **Read hidden properties** and if necessary, **Edit hidden properties**.

Please also note the option to hide individual fields only for d.3 import or d.3 smart explorer.

> **Note**
> By default, hidden properties (hideable) will not be sent to d.3 search. You have to enable this explicit with the parameter `DSEARCH_ADD_HIDDEN_ATTR`.
>
> This parameter must be set manually into the `d3addon.ini` of the repository.
>
> `DSEARCH_ADD_HIDDEN_ATTR=1`
>
> After changing this, you must restart all d.3 processes (`d.3 hostimp`, `d.3 async` and `d.3 server`) for this respective repository to enable the settings.

Advanced

The tab **Extended** allows settings for the conversion and display in d.3 import and the result list or the search form of d.3 smart explorer.

For each property, you can decide whether the values are to be displayed in upper-case, lower-case or in the original.

For category specific property fields in d.3 import or in d.3 smart explorer (search and result list), you can determine the display option in the section **Visible in**. If you disable an option, the property is no longer displayed in the feature.

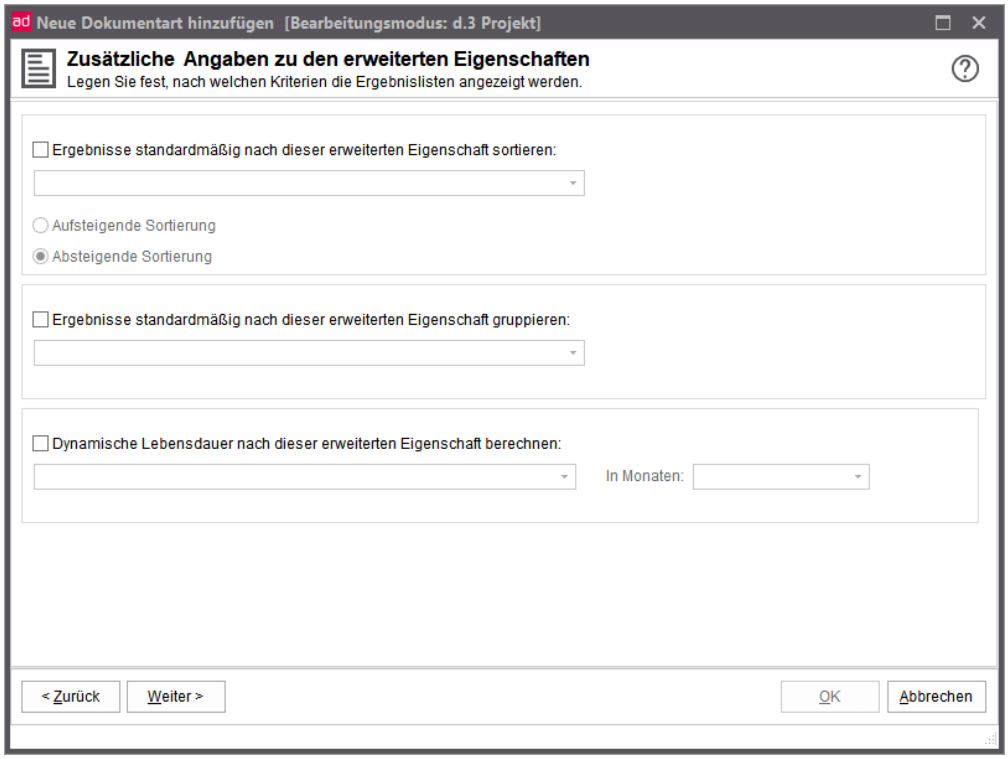
> **Warning**
> Please note that this setting is not equivalent to the global configuration via the property **Visible in**Hideable in connection with the class-based but only supplements it.
>
> If you disable the visibility for a property, a user can still display the property again. The property is still displayed in d.3 smart explorer under **Result list > Arrange columns** in **Available columns**.
>
> If you enable **Hideable** for a property, the user cannot display the property. The setting **Hideable** then affects d.3 import as well as d.3 smart explorer with its advanced search and result list.

Perform a search for the document type (category). In d.3 smart explorer, the d.3 users can display a property, which you have disabled under **Visible in** by doing as follows:

1. Select the context action **Result list** and then **Arrange columns**.
2. Under **Available columns** choose the properties which you want to display in the result list.

Additional information for the advanced properties

You can also preset the display of the results list for document types.



By default, sort results by this advanced property:

This default setting can be "overwritten" in d.3 smart explorer via the user action.

By default, results are grouped by this advanced property

Also this setting is only a default setting and can be changed in d.3 smart explorer in each result list.

Calculate dynamic time to live by this advanced property: This setting allows you to specify that a document of this document type receives a calculated deletion date based on a date in the advanced property (Date fields only). The entry at "In months" indicates when the document should be deleted depending on the value of the advanced property. Besides the selection, free entries are also possible. Negative values and "0" are not allowed. If a value in the past is entered in the advanced property, the document will still not be deleted for a month.

Multiple languages

The multi-lingual properties include the names of the document types, the advanced properties, the group names and the values of static datasets. These names should always be translated, if you intend to support several languages in your d.3 repository.

Let us suppose you usually operate your d.3 system in English and German. If you want to provide your users with more languages, then the product d.3 national must be installed, e.g. for the language Dutch.

To display the property names, document type names or the group names e.g. in Dutch, enter the tanslations in this dialog.

Currently, the default languages German and English are supported in d.3 admin without any further installation.

Add language

To add a language such as Spanish, do the following:

1. Click the button **Add language**. A new dialog window is opened.

2. Select the respective language.
3. Click **OK**.

To remove a language, click the button **Remove language**. The languages English and German are not removable. For all further languages, an entry is created in the file `d3admin.ini` in the configuration directory of the repository.

> **Warning**
> Do NOT edit the `d3admin.ini` manually because it will be written by the server from d.3 version 8! Immediately changes of the file will be lead to discrepancies of d.3 admin and the transport system.

Translating

Once you added all desired languages you can start your translation.

To do so, either select

- Document types
- Repository fields
- Activity profiles

or

- Datasets

Depending on the selection a window is displayed where you can enter the desired translations.

For example, to translate the name of a document type, simply choose the button **Document types**.

Afterwards, you can translate every name with a double-click on the respective row.

When translating the user groups, the respective types are again represented by the icons.

 Group without mailbox-distribution

 Group with mailbox-distribution

 Members of a group with workload balancing

 Organizational unit

Note
In the same way you can translate the advanced properties and the group names.

1. For this effect, choose the pages with the buttons.
2. To return to the start window for the translation of the multi-language properties select the option **Information**.

To translate the static datasets, do the following:

1. Choose the dataset first.
2. Translate into the selected language.

In contrast to the translation of repository fields and document types, the translation is not stored before you confirm it with a click on the button **Apply**.

With **Delete Changes** you can clear all changes not yet applied.

Please note that the advanced property fields accessing a dataset are always stored in the database under the ID. The translations are only determined for the search and for display.

However, you can also translate the static dataset IDs into the respective target language (see Manually created datasets).

Dossier generation

You can find detailled information on the dossier generation in the manual for d.3 admin folder scheme.

d.3 explorer folder browser Admin

For the configuration of d.3 explorer folder browsers read the respective manual.

Automatic storage

To facilitate the storage of multiple documents for the users, it can be enabled that the documents to be stored can be dragged and dropped onto the “target dossier" (d.3 smart start or d.3 smart explorer). Afterwards, all documents are stored in the d.3 repository without any further user interaction. A progress bar informs the user about the running import.

If mandatory fields are defined in the storage document type, which cannot be filled from the properties of the dossier, then the d.3 import application is displayed. All changes applied here are then valid for all further documents to be stored.

If the user has no write access to the specified document type, he receives an according error message and can change the properties to allow a storage. These properties are then valid for all other documents to be stored.

This functionality requires a set of rules which are configured in this plugin.

This set of rules basically describes the following:

If documents are dragged on the dossier of a dossier type listed here, then store the document into the assigned document type with the properties of the dossier.

Rules are defined in the following dialog.

1. To create a new mapping, click on **New**.

2. Select the dossier for which you want to define a mapping rule in the upper selection.
3. Choose the document type later to adopt the properties of the dossier in the lower dropdown list.

If you defined several mappings, then you can manage them with the respective buttons.

1. For this effect, select the mapping.
2. Click on the required button.

Remove: Remove mapping (without confirmation!)

Edit: Map a different document type

Close: Close application

Do not forget to save your configuration with **OK**.

1.4.3. Permissions

In the following, you will find the individual chapters dealing with the topic of permissions.

User

Below is not only general information about user permissions, but also how to create, search, configure and remove users.

User common

Every person, which should have access to the d.3 repository, must first be configured as a d.3 user. The users "batchimp", "hostimp", "d3_server" and "d3_async" are system users and are not displayed in the user overview of d.3 admin.

> **Note**
> In this chapter you can find information on the local user creation and management in the d.3 repository.
>
> If a connection via LDAP is realized, then the users are created in the d.3 repository with their first login.

A further system user is the "d3_admin". It is not necessary to administrate this user. You do not need to assign any authorization profiles, classes or groups to this user.

> **Warning**
> The user name cannot be changed in the d.3 environment. Alternatively, you can create a new user under a different name and copy all files from the processing, the mailbox or resubmissions from the old username into the new user's directories. Once the old user has no documents in his directory structure any more it can be deleted.

In the overview, you can also see, if a user account is blocked. The context menu allows you to unblock the user. The system differentiates between a user which is blocked due to failed login attempts with wrong passwords etc. or, if the it was locked through policies (group membership) (see d.3 policy manager). In this case, the unblocking must be explicitly confirmed.

> **Warning**
> Do not create a "dummy" user, because this user name is already used internally by the server processes.

Filtering and searching of users

The filter and search mechanism greatly simplifies administration, especially in an environment with many users.

In the upper part of the overview window of the user administration you can control the filter and search mechanism.

Search for: Initially specify the search pattern, the string will be searched in all fields of the user list and the result will be immediately reduced respectively.

Search in: The defined string can be searched in a column you specify here afterwards.

Count: Indicates the maximum number of listed users; the count can be predefined via the configuration file in d.3 admin.

Search: Executes the search for the string and highlights the rows, that contain the search pattern in the selected column.

Reset: Resets all entries back to default value; the search fields will be deleted, the count will set on the value of the configuration file (`ManyUserFindLimit`).

Configuration of the user filter

You can customize the behaviour of the user filter in the configuration file of d.3 admin. Two methods exist for this effect:

Example of the file d3admin.ini

```
[Performance]
ManyUserSupportEnabled=1
ManyUserLimit=500
ManyUserUpdateCount=500
ManyUserFindLimit=500
```

The parameters have the following meaning or effect:

Section **Performance**:

Parameters

- **ManyUserSupportEnabled**: Enabling of the user filter for user assignments, too, e.g. for authorization profiles, groups. Values: 1 (default: yes), 0
- **ManyUserLimit**: Count of users, that enables the advanced search. Value: 500.
- **ManyUserUpdateCount**: Count of users to be displayed, after which the display is updated. Value: 500.

- **ManyUserFindLimit**: Maximum count of users while searching via filter. Value: 500.

Create user

When defining a new local user, various data must be entered. In the following, the sequence of dialog to define a new user is described.

> **Note**
> If a connection via an LDAP server is realized, then the users are created in the d.3 system with their first login. Prior to d.3 version 8.0.0 it was erroneously possible to edit a user synchronized via LDAP although these changes were lost with the next synchronization. With version 8.0 you can now no longer fully edit such a user.

Complete Name

Here, you can specify the full name of the d.3 user.

This name is often used in d.3 applications, e.g. for the mailbox mechanism of d.3. This field is limited to 50 characters. The name is more descriptive than the technical name since this is truncated to 10 digits.

The full name is displayed on the title bar of d.3 smart explorer, but also in various picklists.

Department

To differentiate the users, you can assign them to certain departments. The picklist containing the departments is dynamically extended. This means, when you create the first user in d.3, this picklist is still empty. Creating the user n, the different departments are displayed for selection which you already assigned to the previous users. This field is limited to 50 characters.

Organization

The plant is another field to differentiate between users. This picklist is also dynamically extended. The field must not exceed 50 characters.

Phone

The phone number is limited to 40 characters.

E-mail

The user's e-mail address is limited to 100 characters.

If the SMTP support for the d.3 repository is enabled, then an automatically generated e-mail will be sent by the d.3 server. This means that with receiving a document in the d.3 mailbox of a d.3 user an e-mail is generated automatically. Thus, the e-mail system remains the leading system for the distribution of information.

If the subscription service of d.3 is used, then the notifications on changed documents are also sent to this e-mail address.

User name

Enter the d.3 user name here. This is the name used when the d.3 user logs in the d.3 repository. You can use a maximum of 30 characters for a user password. The characters must be taken from the following set ['a'..'z','A'..'Z','0'..'9','_'] . All other characters must not be used in user names.

All characters are allowed, if the d.3 repository supports Unicode. d.3 creates a unique 8-digit name, which contains no special characters.

In the directory `edit` in the document tree of the d.3 repository, a folder is created under the technical name for every user (8-digits).

> **Warning**
> User and group names are significant in the first 10 characters, so, each user and group name must differ from any other user and group name in the first 10 characters. If more than 10 characters were specified for a user name, then the name is truncated to 10 characters by the d.3 system and defined as unique. The original name can be used as an alias for the login.

Example: The user name ClausChef is substituted by ClausCh1. Under this name he is entered in the user administration.

It is not possible in d.3 admin to create a user and a group with the same name. If you receive a respective message, please check if there is a user with this user name already.

Password/Password confirmation

When creating a user, a password must now be specified. Even when changing a password, a new password must be specified. The syntax depends on a variety of configuration parameters.

The parameter `PASSWORD_MIN_CHARS` specifies the minimum Length of a password. It can be defined in the d.3 config.

The parameter `PASSWORT_EXTENDED_SYNTAX` defines the syntactic rules for passwords. If this parameter is set to "Yes", then the password must at least include one sign of 3 of the following 4 character groups: A-Z, a-z, 0-9, special characters.

The option `PASSWORD_EXPIRE_DAYS` specifies the default lifetime of a password in days, but only if the option `PASSWORD_EXTENDED_SYNTAX` is set to "Yes".

The default setting for this parameter is "`10957`", which is equivalent to a time of 30 years.

The password will not be displayed, when you edit a user. Thus, the fields password and password confirmation remain empty. Hence, changes to a user's settings doe not affect the user's password. The old password is overwritten only if the new password is explicitly specified.

You can use a maximum of 20 characters for a user password. The length and syntax rules for a password can be further configured in the d.3 configuration.

If the password does not comply with the valid rules then an according message is displayed.

Advanced properties - User rights

The advanced properties (policies) are further described on the chapter d.3 policy manager. Thus, their configuration will only be briefly mentioned here.

For every user rights can be given for the listed properties. The assignment here has the same meaning like the rights in the authorization profiles.

Assignment

yes: Right assigned

no: Right denied

<Empty> Ignore right

The rights are assigned either with a click on the respective button or with a click on the property. Every click changes the property value in a cycle.

The user is enabled to log in via d.3 login manager. However, if the user is configured as a service-user, then it cannot log in from the dialog (implicit denial).

You can define rights both on applications and on documents.

Optional fields

The optional fields allow you to store additional information about a user. You can enter a maximum of 50 characters in each of the 10 optional fields. In older d.3 systems the optional fields were often used to store the e-mail address.

The superior of a user must then be specified in an optional field of the user. You can also use the optional fields to store the superior of a user so that he can change his checkout-status in case of sick-leave.

Also d.3 abo service reads an optional field. The optional field 1 can contain the ID of the abo-manager (see documentation d.3 abo service).

The caption texts cannot be changed.

Group membership

Every user can be member of multiple groups. User groups combine employees with the same job profiles and identical rights. More details can be found in chapter Groups.

> **Warning**
> Please note that the assignment process can take several minutes if you select more than 10 groups.

Here, the symbols and text labels have the following meaning:

 Distribution of documents to all group members (mailbox without workload balancer)

 Distribution to one member of the group (mailbox with load balancing)

 no distribution, the group is used to assign permissions

Authorization profiles

With the concept of access rights in d.3 you can directly assign rights via an authorization profile.

Here, you can assign one or more authorization profiles to a user.

1. Choose the authorization profiles on the right.
2. Add them to left window with the button **Add**.

To remove an authorization profile, do the following:

1. Select one from the left window vice versa.
2. Click the button **>**.

The new user will be created, as soon as you click on **OK**.

Assign document classes

Additionally or alternatively, you can also directly assign document classes to a user. This way, you could extend the rights, e.g. for key users or further restrict them.

Overview of user data

1. Open the user overview via **Rights | User**.
2. Choose a user.
3. Choose the **Properties** with the context menu option.

Then open the properties to see an overview over the selected values of the user.

Über die jeweilige Tab General, Assignments, Advanced properties) können Sie die konfigurierten Daten einsehen und auch edit.

General

On the tab **Common** you can see and edit the **User data**, enter **optional fields** and leave a **Checkout message**.

Reset password

Via the button **Reset password** you can define a new password for this user for the d.3 login.

Unbind LDAP

The button **Unbind LDAP** is available, if the user is a LDAP-user and allows to unbind LDAP.

Move documents

If you want to move all documents or document type of user to another or to remove it, use the dialog **Move documents**, which you can open by click on the button with the same name.

In the left half of the window you can see **Information to the user** and the **document types**, in which the user owns documents. For this effect the amount of **not adopted documents** per document type is visible in each category.

In the right half you can choose a **user or a group for adopting the documents**. For the user or the group the same information will be displayed. In addition to that, for the user are all document types displayed, on which he has the required rights. For adopting you can choose between documents from the **mailbox** or documents in **Processing**. The **documents to be adopted** will be displayed below.

To move documents the following steps are to be done:

1. Choose a category from which the documents are to be moved (mailbox, processing).
2. Check the documents of the desired document types to be moved in the list.
3. Click on **Apply** to move the documents or
4. on **Reset** to discard the changes.

> **Note**
> Moving documnets is not without risk and should be done with care!
>
> For users mailbox entries and documents in processing can be moved.
>
> Removing entries is an additional opportunity for mailbox entries and job profile links. The documents must be adopted previously.

Assignments

On the tab **assignment** allows you to see and edit the assignments of the user in **groups**, **authorization profiles** and **document classes**.

Advanced properties

The tab **Advanced properties** shows the respective rights of the user, which you can manage via the buttons **Assign right**, **Deny right** and **Ignore right**, see Advanced properties - User rights.

Remove user

To remove a user, do the following:

1. Choose the respective user from the **Overview of Users**.
2. Click on the button **Remove**.
3. Confirm the following query with **Yes**.

If the user to be removed has still documents in processing in his mailbox, an additional dialog appears, which allows you to move these documents to another user or to delete these documents, too.

Licensing

Concurrent user

In this model several users share one license.

If a user executes a function in the d.3 repository, then a free license is occupied by him/her for 30 minutes. If the same user executes within the 30 minutes more actions, the occupation is set to 30 minutes for each action. If none actions are made within the 30 minutes, then the license can be used by another user after this time.

The login to d.3 smart explorer / d.3 smart start does not occupy a license.

> **Warning**
> From version 8.0 of d.3 smart explorer the license is not being released directly after closing the application. The license is available after 30 Minutes.

Named Users

In this model each user occupies a license. For each user who shall work with the system, a license is required. This also applies for part-time employees.

General information

With d.3 version 8.1 the new license typ d.3ecm user S was introduced. Users of this license type only have read-only access to the document in the d.3 repository and are allowed to participate to a workflow.

For this license type the d.3 server differentiates between an "only reading" user (called read-only-user) and a "writing" user (called full-user).

Reading the license data

When starting the d.3 processes the following license types are read from the `liccheck.lic`:

- "d.3-named" for full-user
- "d.3-named read only" for read-only-user

In the d.3 log this is indicated by the log entries:

- "Max. number of full users = ..." and
- "Max. number of read-only users =...",

System group "$ReadOnly$"

A new system group "$ReadOnly$" was introduced and by the membership of a user either a full-user license (not member of this group) or a read-only-user license (member of this group) is occupied.

- The group is only created, if really read-only-user licenses are accounted in the license file.
- The group cannot be changed or deleted.
- Only the user memberships can be changed.
- Service users cannot be a member of this group. Neither is it possible to adopt other groups as members of this group. This group also cannot be member of another group.

Read-only users

A user occupying a read-only user license has only read-only access to documents. A document import, changes to document properties or linking documents is therefore not possible for these users.

Phase after replacing the license file

If a named user license was found in the license file after the first start of the d.3 processes, the phasing-in period of 30 days begins. In this phase the users can switch between full-user license and read-only-user license without time delay or blocking the licenses.

Because all d.3 users are full-users at first, it is necessary to assign the users to be act as read-only users to the group "$ReadOnly$" in d.3 admin after the installation.

Occupancy of a license

A Named User license is only occupied when a user actively logs on to the d.3 system. No licensing takes place during user creation. In the case of user changes, especially those involving a license change, only

the previously occupied license is deregistered with the license server and blocked for 5 days after the introductory period. The new license is not registered.

Daily registration of licenses

The async process runs through a routine once a day to register users with the license server. The following rules apply for this:

- Disabled user accounts are not registered.
- The registration distinguishes between the read-only and full-license.

Switching a user to a read-only user license

Within the phasing-in period of 30 days of the named user license users can be changed to read-only users without blocking a license.

Else the following rules apply:

- The restricted rights for the users take effect immediately.
- After the introductory period has expired, the previously occupied full-license is blocked for another 5 days.
- The read-only license will be occupied by the async process at the next login or the next daily registration.

Switching a user to a full-user license

A previously occupied read-only user license is immediately released. The full-license will be occupied by the async process at the next login or the next daily registration.

Service users

A service user occupies no named user license. But if the service user becomes a "normal" user, it occupies a license that becomes active at the next login or through the daily registration by the async process.

If a named user becomes a service user, after the introductory period the respective license (full-user or read-only user) is blocked for 5 days.

Deleting users

After the deletion of a user the previously occupied license is blocked for 5 days.

Disabling users

If a user has been disabled, the occupied license is blocked for 5 days.

License management with an LDAP connection

- Each new LDAP user initially occupies no license.
- By an assignment to the system group "$ReadOnly$" via the LDAP configuration the user occupies a read-only user license.
- The behavior on license switches is the same as without LDAP connection: When switching a license to a license with a smaller range of functions the previous license is still occupied for 5 days.
- Comfort function: If a user loses all rights because no groups or authorization profiles are assigned via the LDAP configuration, the user will be disabled in d.3. This also releases the license of this user after 5 days.
- Comfort function: If a disabled user gets d.3 access rights via the LDAP configuration again, the user will be enabled automatically. The user will occupy a license with the next login.

- If a user still exists in the d.3 repository but not in the directory service, the user will be disabled in d.3. This releases the license (after a block of 5 days due to the license switch).
- An automatic verification checks regularly, if a user has or has not any rights in d.3. It is sufficient to revoke the d.3 rights from the users and the rest will be done by the automatic license release.

> **Warning**
> If the message "The system group '$Deactivate$' no longer exist! Please create a new group with the right 'Account is disabled' and assign it instead." after a login of a user or during the synchronization of the LDAP users by the d.3 async process appears, then do the following:
>
> The reason for this message is that the system group "$Deactivate$" was adapted to the LDAP configuration to disable users and to avoid occupying a license at the same time. Due to the reason that from version 8.1 hotfix 6 of d.3 server disabled users do not occupy a license, the group was deleted entirely. Instead now another group (at best a group with the option "No distribution to group mailbox") needs to be created for which the advanced property "User->d.3->User account->Account is disabled" is set to "Yes". Then, this group needs to be set in the LDAP configuration instead of the group "$Deactivate$". If this recommended change is not applied, then the functionality is given as before in spite of the removal of the group "$Deactivate$".

Groups

Below is not only general information about group permissions, but also how to create and manage groups.

Groups common

Groups are used in d.3 to combine users with the same tasks in order to select them as a recipient in the mailbox dialog or to use them in the concept of access rights.

> **Note**
> The group name can have up to 100 characters.
>
> The group name must differ from any other user or group name in the first 8 characters.

Traditional boundaries for user groups and authorization profiles no longer exist. Nevertheless, keep an eye on main memory usage and performance when evaluating document permissions. In particular, a high number of permission assignments increases the main memory requirements and has a cumulative effect on search speed when the number of results is high.

All available groups are displayed in the group overview screen. Using the respective icon, the type of group is illustrated.

Select a group to see a list of all members in the right window. As you can see, groups can also be members (nested groups).

 Distribution of documents to all group members (mailbox without workload balancer), group type cannot be changed later.

 Distribution to one member of the group (mailbox with workload balancer), group type cannot be changed later.

No distribution, all employees will be grouped (initially), group type can be changed later, if e.g. the type of the mailbox distribution is checked.

To filter for group types, click on the respective icons above the list.

Use the button **Remove** to delete a group. A multiple selection is possible in the group overview, so that several groups can be removed in one step. During this process, however, the members' display is cleared and the buttons **New**, **Edit** and **Copy** are disbaled.

> **Note**
> If documents are still assigned to a group when you delete them, a corresponding dialog is displayed. Here the documents can be moved.
>
> When deleting several groups, this dialog is displayed separately for each group. If the documents are not moved, you can cancel the process for this group by a new query and continue with the next one.
>
> If the process is not cancelled for a group, but the documents are not moved either, you will end up in a kind of endless loop.

Adding a group

With the button **New** on the overview of d.3-groups you create a new group in the d.3 repository.

First, enter the general information.

Type of document distribution:

Distribution to all members (group)

If a document is sent to the d.3 mailbox, then the document can be sent to all group members with the same rights. Every group member receives a notification in his d.3 mailbox.

Group documents should then be accepted by a member of the group for further processing (see d.3 smart explorer manual).

Distribution to one member (activity profile)

If a document is sent to the d.3 mailbox, then the document can also be sent to an job profile. In contrast to the sending to a group, exactly one member of this group receives the document as a task in his mailbox. Assign additional values to the workload balancing later for the control.

No distribution

This group does not appear in the d.3-applications, e.g. in the mailbox-dialog. Here, you group members only. The type can be changed later. When creating the company structure, you can also use this group.

Creating user groups - extended properties

Additional information can be found in the advanced properties at the user properties or the chapter d.3 policy manager.

Mitglieder

Members of a group can be users as well as user groups.

Choose the members of the group to be created from the list in the right window. A multiple selection can be applied using the usual Microsoft Windows functionality. You can assign normal mailbox groups only.

Authorization profiles for groups

With d.3 version 7 authorization profiles can now also be assigned to a user group.

Having successfully created a new user group, the following information is displayed.

Load distribution (activity profile)

If a group is configured for the distribution to one member (activity profile), then the members of this group appear in the overview.

For each member the fundamental information with reference to the distribution of tasks can be assigned in the following dialog.

Initially, define the type of load balancing:

Load balancing

Based on the weight and the existing mailbox entries, the mailbox dispatch is controlled. For this effect, the parameters in the d.3 configuration are interpreted.

Fill level distribution

The mailbox dispatch is controlled based on the threshold values. Tasks will be assigned as long as the maximal value is reached. The weighting and the current balancing is considered for this.

Equal distribution

The mailbox dispatch goes by the number of the documents which have been sent previously via this activity profile to the user and not by the current number of the mailbox entries. The weighting is included here.

Configure the weighting for the selected member by double-click.

You can gather or edit the settings for the load balancing to the selected user or group.

Deputy

Enter the employee's deputy here. This delegate can receive documents from his colleague's mailbox in the absence of his colleague via the mechanism of the delegation rules. You can also specify a group as deputy which makes all members automatically a deputy for the employee.

Weight

If you want to implement a conversion that distinguishes between part-time and full-time employees, for example, so that this is taken into account when sending documents, then define a (self-selected) weighting here. Any values are permitted.

Example 1: "8" could be equivalent to an 8-hour day (full-time), while "4" could represent a 4-hour day (part-time).

Example 2: Count of tasks (on the "desktop") of a employee: by reaching the maximum count, he no longer gets a task (in relation to all other group members).

Maximum treshold for document distribution

For the mechanism of the fill level distribution you can enter a freely definable lower limit here, under which an employee shall receive additional mailbox entries again.

Example: Enter the number of mailbox entries. Whether all entries or only the unread ones are counted is controlled by the corresponding parameter in the delegation rules.

> **Note**
> If the upper threshold value is reached for all users, an internally determined coefficient determines the next recipient of a document. It is therefore guaranteed that a document will always find a recipient.

Lower threshold value for document distribution

For the mechanism of the fill level distribution you can enter a freely definable lower limit here, under which an employee shall receive additional mailbox entries again.

Example: Enter the number of mailbox entries here, from whcih the employee should receive additional mailbox entries again.

Head of job profile

If you want the employee to control this section, enable the checkbox. This employee is then eligible to set the colleagues in his activity profile absent or available (see Substitution rules). If he was granted the respective permission, he is also able to view the mailbox entries of his group members.

Participates in document distribution

You can remove employees from the distribution of documents, for example, a possible employment for the superior.

> **Note**
> If a group is selected for the configuration of the load distribution, only the weighting or threshold values can be entered.
>
> Additionally, the group can be excluded from the document distribution.
>
> If different weights or thresholds are defined for the user (directly) and for the group, then those values assigned to the group apply. The user entries are not interpreted in this case. This also applies to the participation in the document distribution.

Group properties

You can get to the group properties after opening the **Overview of groups** (**Rights | Groups**).

- Choose the desired group and click on the button **Edit**.
- Or perform a double-click on the name of the group.

On the tabs **Common**, **Advanced properties**, **Members** and **Authorization profile** you can see the entries, done while Creating the group and change them via **Edit** later.

The button **Move documents** provides to move documents or mailbox entries of a group to another group or user. The function resembles the button with the same name in the Overview of users.

Additionally to the mailbox entries and the documents in processing for groups it is possible to move documents in verification and job profile links.

> **Note**
> In contrast to users for groups only document types from which the group has documents are displayed in the right list of the dialog.

Document classes

Document classes common

Document classes are a part of the rights concept. Document classes allow you to define rights to "documents of a document type".

A simple example of "documents of a document type" is the permission to display "all invoices with an invoice amount of less than 10.000 €".

Document classes always determine a subset of the possible documents.

Basically, there are two types of document classes. The first is the document type dependent class type (here supplier order (K)) while the other one is document type independent (or even cross-document type) class type.

The document type-related document class is easily understood as it defines a sub-set of an entire document type using certain criteria such as the limitation of "all invoices" to "invoices below 10,000 € in the above mentioned example.

In contrast to this, a document type-independent class is used to limit "documents of all document types" in the d.3 repository. These types of document classes are an additional criteria to restrict the access. They cannot be assigned to a user as the only document class.

Let us suppose, you want to grant a user access to a released documents. Then it is not sufficient to generate a **document type-independent** document class with the restrictions "Status = Release", and to assign this to the user. The user would then not see any (concrete) document- and dossier types in d.3 smart explorer or d.3 import.

You rather create a new authorization profile and assign all default classes and the **document type-independent document class** to this authorization profile.

The default classes are automatically generated by the d.3 server with a new document type or dossier type. The default classes are marked with the suffix "(K)" ("C" in English repositories) and a class ID with the prefix "CLAS_".

Afterwards you assign permissions only in the **document-type independent class**. Finally, you could specify a name for this profile (such as "All documents") and finish it.

This authorization profile is assigned to the user. The user will only see only documents in the status Release as the results of a search.

If you want to implement a contrasting right, (in our example: provide a user with all documents not in the status release), then please follow these steps:

You refuse the reading rights for the document type-independent class ("NO " or "right ignored") but you have to grant reading rights for the default classes ("YES" or "right granted"). For a better understanding, see the document- type-independent class as a template with black areas and the other classes as templates with "holes". All templates are places on to of each other. You can then only see through the holes while the black areas are still opaque.

> **Warning**
> The user / class assignments should only be used together with authorization profiles, if you explicitly want to grant or deny certain rights.

Add document class

1. To add a new document class, click the button **New** in the overview.
2. Initially, select the document type for which you want to define restrictions, e.g. “Supplier Order”.

Definition of restrictions

If you selected a document type to which the document class is to relate, only the properties of this document type are displayed in the table. Then define the restrictions in the individual fields.

Depending on the data type of a field, you can also work with special characters and macros.

Under **Identification** enter a descriptive name for the new document class designating some of its restrictions.

The name of a document class has only informative character. You can enter a maximum of 255 characters. Thus, you can specify easily recognizable telling names. A reasonable name could for example be "Invoices with an invoice amount under 10,000 €".

Additionally, you must specify a max. 10-digit distinct technical name (short name). This name is used in the hook-development. The short name is case-sentive, i.e. upper- and lower-case are interpreted.

All default document classes have short names with the prefix `CLAS_`. This prefix is exclusively reserved for the default document classes. You cannot manually generate a class with this prefix.

Via the option **Adopting filters as a search condition** it can be configured that the restrictions of the document classes can be integrated into the SELECT statement for the search for documents.

Consider the following for this effect:

- Only the document classes are considered which are assigned to the user via the authorization profiles or directly via document class assignments.
- If a search is executed for exactly one document type, the restrictions of this document classes based on the created document type apply.
- If a search is executed for none or several document types, only document type-independent document classes will be read.

- Only the restriction of one document class can be integrated in the search criteria because the document classes interpretation is always executed via OR-connection and this can not be implemented in SQL-commandos, especially when multiple restrictions per class are set. This means: If multiple document classes are assigned with option to the user and these document classes refer to the same document type, then none document class will be considered.
- For the integration of document classes restrictions only the properties in the search criteria will be set, which have not been set by the user already.
- The following macros are supported: @D3USER, @D3GROUP, @D3USER_REALNAME, @D3USER_LONG, @D3USER_EMAIL, @D3USER_LDAPDN, @D3USER_OPTFIELD, @D3SET

Example:

A document class of the document type "e-mail" is specified (document type short name EMAIL) with the restriction "@D3USER_EMAIL" to the property field "recipient" (DB-position 1).

The user "user1" has the following entry as e-mail address: user1@firma.de

When searching for documents in the document type "e-mail" without more search criteria the following SQL-command is generated (abridged):

SELECT * FROM firmen_spezifisch WHERE kue_dokuart = 'EMAIL' AND dok_dat_feld_1 = 'user1@firma.de'

Restrictions

A document type can be restricted with a restriction, similar to a search in d.3. Hence, the usual wildcards can be applied in the restriction. The possible wildcards depend on the data type of the field. Please note the advanced options for alphanumeric special characters.

Alphanumeric fields

You can use wildcards in text fields defined in the d.3 configuration. By default these are the "%" (percent) and "_" (underscore). The % represents an unknown number of characters and the underscore is a placeholder for exactly one character.

Let us suppose, your d.3 repository contains a document type called "Delivery notes". This document type holds a field named "Supplier". This alpha-numeric field should hold the name of the suppliers. In order to allow a user to access documents relating to the supplier “d.velop AG”, you would have to enter the restriction “d.velop AG” in the respective field.

The disadvantage of this exact designation would be that documents filed with a supplier like “Fa. d.velop AG" would not be included. At this point it would be preferable to enter “%d.velop AG”. Now, both spellings would be found. If some documents were stored with an entry like "Fa. d.velop", then again this would not be included in the results for this restriction. To avoid this, you would have to restrict with *d.velop*. The "." in the company name could pose a final problem. Documents where the supplier was specified with a "-" (hyphen) such as "d-velop AG" are still not found. The remedy would be the pattern-matching character "Underscore” (“_”). Thus, the final restriction for our field "Supplier" in the document type "Delivery notes" would be %d_velop%. This value must be entered when defining the document class.

Note

You can also use alternative placeholder characters for "%" and "_" such as the equivalent "*" and "?". From d.3 version 6.2 it is also possible to define additional special characters for the alphanumeric search.

These individually defined characters can be used for the search as well as for the definition of document classes.

However, for document classes, you cannot combine the special characters with the exception of "*", "%","_" and "?".

Exception: Combinations of "*", "%","_" and "?".

Date fields

You can use the minus sign ("-") in date fields. This allows the definition of ranges "from – to", "since –" and "- until".

Ranges	Restrictions
from - to	01.01.1997 - 31.12.2001
from...	01.01.1997
to	- 31.12.1999
{-m}-{+n}	Example: {-28}-{+28} restriction for the range from (4 weeks ago) until (in 4 weeks)

Let us suppose, your d.3 repository contains a document type called “Invoice". This document type includes an property named "Date", in which the invoice date is stored.

If an user should have access to all invoices up to the "31.12.2000", you would have to create a document class using the document type "Invoice".

Furthermore, you have to restrict the field **Date**, entering "-31.12.2000". Finally, this new class would have to be assigned to this user.

Numeric fields

You can use the "–" (hyphen) in numeric fields to designate a range.

An example could be the "invoice amount". If you wanted to create a document class for the invoices of less than 10,000 € then you would enter the restriction "-10000" (-up to). In order to specify invoices with an invoice amount larger than 10,000 € you would enter "10000-"(from-).

To search for negative numbers, you would have to precede the number with a back-slash "\".

Field **deviation in mm (longer / shorter)** from the standard when producing a good.

- 100-
  All deviations where the produced good is more than 100 mm longer than the standard-good.
- -10
  All deviations where the produced good is more than 10 mm shorter than the standard-good.
- \-10
  Search for goods being exactly 10 mm shorter than the standard-good
- 10-100
  Search for goods being at least 10 mm longer and up to 100 mm longer than the standard-good

Dataset restrictions

When defining a document class the values for a property with a dataset, its values are displayed for selection via the right combo box.

The stored dataset is displayed.

Technical fields

The upper section of the overview displays the technical fields available as restrictions. These fields are document type-independent.

Document number: This alphanumeric field represents the **drawing number** (`zeich_nr`).

Document ID: The document ID is a unique identifier for the document in the d.3 repository. This field is alphanumeric.

Created on: In this field you can specify the restrictions for the creation date of a document. It is a date field and the restrictions have to be applied as described before.

Released/Blocked: This property affects released documents. When defining a document class, you can either enter "Released", "Blocked" or nothing in this field.

The values "Released" and "Blocked" are shown via clicking the combo box on the right side. The combo box appears by clicking two times into the restriction field.

Let us suppose, a user should gain access to all released documents of the document type "Manuals" then it would be enough to set the status (compare chapter Status) with the value "Release" as a restriction (Please note the special case in the chapter Status). Thus, the user has access to all released documents of this document type, even to the blocked documents. To prevent the user from accessing the blocked documents, you must fill the field **Released/Blocked** with the restriction "Released".

> **Note**
> Blocking a released document is useful if it proves to be faulty in content or its validity has expired.

Plan/Verified: This field applies to the first step of verification. The allowed values for this field are “Yes", "No" or nothing. If you select "Yes" here, the respective document must have been verified already. The values "Yes" and "No" are shown via clicking the combo box on the right side. The combo box appears by clicking two times into the restriction field.

Status: This field refers to the status of a document. Possible values are "Processing", "Verification", "Release", "Archive" and nothing.

> **Note**
> When the permission is determined, always the most recent status is checked. This means that a restriction like "Release" will only return those documents with the current status "Release". All documents which are additionally in "Processing" or "Verification" are ignored.
>
> In order to define a restriction for "all released documents", leave this property empty and enter the string "Released" in the field **Released/Blocked**.

Editor: This field applies to the first step of “verification”. You can either select a user name or a group name from this picklist.

You can also enter a macro (see chapter), especially @D3USER or @D3GROUP Macros).

This property is an alphanumeric field.

Web published: You can publish documents for the web. If a document was published for web-access, a little globe icon is displayed in the d.3 smart explorer. To grant a user access only to these web-published documents, you must choose the value **Web Published** from the list. In order to explicitly permit access to the documents NOT published on the web, choose **NOT Web Published**. The third option is to select a NULL value. In this case the property **Web Published/Unprotected Web Access** has no influence on the permissions.

> **Note**
> From d.3 version 6.3, this property is called **Unprotected Web Access**. However, it provides an obsolete function since it is based on CGI scripts and the d.3 Web Publisher will no longer be supplied in the future.

Owner: In this field a restriction to the owner of the document can be created. The owner is the user name the document imported.

Client ID: 'There is the option to define a client ID for restrictions. Every application is equipped with an internal identification. This is a 3-digit code, for example "200" for d.3 smart explorer. This way it can be made sure in the future that new or external applications can be allowed or rejected via the rights concept. These settings are currently mainly used for customer-specific solutions. For example, d.3 smart explorer, d.3 import and d.3 view are used with the following IDs (200, 220, 100). The IDs are transmitted to the d.3 server with every API-Call and can be found in d.3 logviewer.

File extension: From d.3 version 6.3 a restriction using the file extension is possible. This way it is possible to deposit the users just determined data types.

Macros

The following macros can be used when defining the restrictions:

@D3USER

If you enter the macro @D3USER in a field as a restriction for a document class, then this "placeholder" will be filled with the current d.3 user name, when the permissions are defined.

Let us suppose a document type "e-mail" is configured in your repository. This document type holds a field named "user". When importing a document, this field is automatically filled with the d.3 user. All users of the d.3 repository get access to zjos document type. The access to this document type is controlled by a document class. This document class contains the macro @D3USER as a restriction for field "User". If this user is searching for e-mails stored in the d.3 repository using d.3 smart explorer he will only see the documents containing his d.3 user name in the field "User".

@D3GROUP

When using the macro @D3GROUP it will be checked during runtime, whether the respective property contains a d.3 group name, of which the user is a member. If this is the case, the user will be granted access to the documents.

@D3USER_OPTFIELD_XX

In order to decide whether a document is included in a class or not it can be helpful to use the macro @D3USER_OPTFIELD_xx in a property field. The effect is that the property is checked against the content of the optional field xx (1 to 10) of the active user from the user administration.

The optional field five in d.3 admin of user "John" contains the value "23" for the department number. You define a class where the property field "Department number" is restricted by the macro @D3USER_OPTFIELD_05. The user will only see those documents where the property field "Department number" equals "his department" (23). If this matches, the user "John" has access to this document.

@D3USER_LDAPDN

The document class macro @D3USER_LDAPDN is used during the rights definition for a document and replaced with the LDAP Distinguished Name assigned to the d.3 user the when the LDAP support is enabled.

When using the Microsoft Windows Active Directory Services (ADS) as an LDAP server this is the ADS-GUID of the respective Microsoft Windows user.

> **Note**
> For the interpretation of the macro a character string "GUID" is removed for a prefixed ADS-GUID.

@D3HOOK

> **Note**
> Since d.3 version 6.2, the option to define and maintain restriction sets provides a way to do away with various small hook functions. Before you consider a solution with a hook function, you should always check, if a solution with a restriction set is possible (see chapter d.3 restriction set manager).

The use of the macro @D3HOOK allows for the development of a hook function to decide on the permission to access a document. The macro receives the name of the hook function as a parameter, e.g. @D3HOOK (procedure). When the right is evaluated (i.e. this function is called) the property value of the document is passed to the hook function. If the hook function returns the value "1" (TRUE), then the condition is fulfilled and access is granted. If the functions returns a "0" (FALSE), the condition is not fulfilled and the user is not permitted to access the document.

> **Note**
> A solution comparing a hook function and restriction sets:
>
> As has been described before for numeric fields, you can only use the restrictions "FROM-TO", "-TO" und "FROM-". A check for several ranges is therefore not possible.
>
> Let us suppose you want to check a numeric field. The user should be permitted to access the document if the numeric field (in this example: "Sup_Nr") is between "1 and 10", exactly 15 or larger than 100.

Problem solution with hook function

A hook function for these task could look like this:

```
proc  liefer_nr_klasse (liefer_nr)
  {
   if ( liefer_nr >= 1 && liefer_nr <= 10 \
       || liefer_nr == 15                 \
       || liefer_nr > 100                 \
      )
      return (1)
   else
      return (0)
  }
```

The macro would have to be specified as follows: @D3HOOK(suppl_nr_class).

Problem solution with a set for numeric values, name: suppl_nr

1-10

15

100-

Problem solution with a set for alphanumeric values: suppl_nr

1~10

15

>100

Requirement for this example: Definition of the extended characters for the alphanumeric search (~ and >)

Referencing the set in the document class definition: @D3SET(suppl_nr)

> **Note**
> Before developing your own hook functions you are urgently requested to attend a d.3 hook training at d.velop AG.

@D3SET

This macro allows to assign a defined restriction set to an property field. With the button **...** you can choose from the defined restriction sets. Please make sure that the data types are suitable.

Afterwards, the restriction set is referenced via the macro. Information on creating a restriction set can be found in the chapter d.3 restriction set manager.

@D3REGEXPR

This macro allows to use regular expressions for restrictions. The possible syntax will depend on the choice of the "interpreter" of the regular expressions (compare the information about the parameter **REGEX_SYNTAX**).

This way basic restrictions can be created without using a hook function or a restriction set.

Example for a restriction to the character range A-H.

@D3REGEXPR([A-H].*)

[A-H]

Exactly 1 character from the range A to H

.

Any character.

*

Repetition of the previous regular expression (here: any character) 0 to n times.

For further information and examples on regular expressions compare the chapter Regular Expression.

@D3USER_REALNAME

This macro is substituted by the real name of the user as specified in the user properties.

@D3USER_LONG
This macro is substituted by the login name of the user as specified in the user properties.

@D3SUP
With this macro the "calling" user is checked for being the disciplinarian of the entered username in the advanced property field.

@D3USER_EMAIL
This macro is replaced with the email address of the logged in user.

@D3IDPUSER
If you enter the macro @D3IDPUSER in a field as a restriction for a document class, then this "placeholder" will be filled with the current ID of the Identity Provider (IDP), when the permissions are defined.

Let us suppose a document type "e-mail" is configured in your repository. This document type holds a field named "user". When importing a document, this field is automatically filled with the IDP ID of users. All users of the d.3 repository get access to zjos document type. The access to this document type is controlled by a document class. This document class contains the macro @D3IDPUSER as a restriction for field "User". If this user is searching for e-mails stored in the d.3 repository using d.3 smart explorer he will only see the documents containing his IDP ID in the field "User".

You can only use this macro if you log in to the d.3 system via Identity Provider.

@D3IDPGROUP
When using the macro `@D3IDPGROUP` it will be checked during runtime, whether the respective property contains a IDP ID of a group of which the user is a member. If this is the case, the user will be granted access to the documents.

The information in which IDP groups a user is a member is determined at login and stored using jStore. This cached data has a validity of 30 minutes (if no credentials to other d.ecs apps are set using DECS_SERVICE_USER, the validity is 10 days).

Changes to a user's IDP group memberships can be captured by the @D3IDPGROUP macro as only when

- if the IDP already delivers them (attention: The identity provider may not provide this directly due to its own cache).
- the user logs in again
- or, when the expiration time of the jStore cache entry is reached

Document type independent class
In contrast to that by choosing **document type independent** all database fields (DOCUMENT_FIELD_1 to DOCUMENT_FIELD_89) are displayed under the name DB Position X. Thus, the restriction applied here does not logically apply to an property (e.g. Supplier name) but physically to a table field.

Please note for your planning that the respective document class to which a document type independent class should apply has identical database positions. This might be the time, when the meaning of the "Preferred position" in the advanced property field creation becomes clear.

If you choose to edit or create a “**document type-independent**" class, all possible property fields (DOC_DAT_ FIELDS) are displayed.

Under the label "DB Position 1" you have to understand the DOC_DAT_FIELD_1. In these property fields you can use the above mentioned special characters as restrictions depending on the data type of a field, as well. The positions

- 1 to 49 are text fields,

- 50 to 59 are date fields,
- 60 to 69 are multi-value property fields,
- 70 to 79 are currency (numeric) fields and
- 80 to 89 are numeric.

The above mentioned options are also available here.

Complete overview of document classes.

The complete overview provides an outline of all document classes. If you want to save the complete overview, do the following:

1. Click the button **Save**.
2. Or press the shortcut key **Ctrl+S**.

The short name (ID) of a document class (first displayed column) is assigned by d.3 server. You have no influence on the numbering.

d.3 allows the adhoc-inheritance of documents. For its implementation, document classes for the respective document are generated. Since these document classes should not and cannot be edited, they are not displayed in ordinary dialogues. However, the **Complete overview** shows them specially highlighted.

You can find further information on this in the chapter d.3 inherit rights manager.

Class usage

With the button **Usage** on the overview of document classes you can see in the filter when a selection is defined, in which authorization profiles this document class is used.

Authorization profiles

Authorization profiles common

> **Note**
> In previous versions, the authorization profiles were called "roles". The name is partially still in use internally.

An authorization profile is a combination of document classes and can be assigned to a user or a user group in order to determine the rights to access documents.

The members of the d.3 project team decide for all roles (departments, responsibilities, etc.) which documents should be accessible according to their range of tasks or job profile.

An authorization profile can now be implemented as the counterpart in d.3 so that this authorization profile just has to be assigned to the members of the department such as “Apprentices Purchasing§ or "Sales Local Office Southern Europe“.

Your company already uses the document types "Delivery Note", "PurchaseOrder" and "Invoice". Your employees in the purchasing department should be able to read these document types. To realize this, you could create three document classes with the permission to read for these document types.

Then, you would assign these classes to each user in the purchasing department with a direct user/class assignment. Now it could happen that sometime later a new document type is needed in your company called "Reminder". Again, you would have to create a new document class for the rights to access this document type and assign it to the users again. This would be time consuming.

The better solution would be the creation of an authorization profile. Here, too, you must create the three document classes for the document types. Then you create an authorization profile with a name like "Purchase" and assign the three document classes to it. Then, all users from Purchasing are assigned this authorization profile (user/ authorization profile assignment). If you now create a document type "Payment order" four weeks later, then it is sufficient to include the newly created document type into the previously created authorization profile.

Warning
The name of an authorization profile must not contain any special characters except "+", "-" and "_". Other special characters are not permitted.

Selection of document classes

The first step in creating an authorization profile is to select the appropriate document classes. The number of document classes assigned to a role is not limited.

From the list of the available document classes, select the ones intended for the authorization profile.

Since the selected document classes may include so called document type independent ones, you can optionally limit the global restriction to this profile.

This is due to the fact that for such global document classes, the database table is only interpreted based on a database column. If this database column was however used for different values in other document types, this would result in an incorrect interpretation.

This can be the case, for example, if you assign several profiles to employees. The document type-independent class would be placed over all document classes contained in the profiles as a "template".

Rights assignment

The excerpt shows the vast variety of options for the assignment of rights. For example, you can now prohibit a user to create notes or redlining elements via profiles and documents classes. Moreover you can prevent a user from changing the status to "Archive".

The buttons below the overview of rights have the following function:

Assign right: The authorization is always assigned for the selected right.

Deny right: The authorization is always not assigned for the selected right.

Ignore right: The authorization is neither granted nor denied at this point. The system checks whether the authorization is granted in another authorization profile assigned to the user. If this is not the case, the authorization shall be deemed not to have been granted.

On the right side you can also select several entries at once. Alternatively you can set the authorizations via keyboard shortcut:

- **CTRL+1**: Assign right
- **CTRL+2**: Deny right
- **CTRL+3**: Ignore right

> **Note**
> If you want to select several entries at once and set a right for them, use these keyboard shortcuts. The buttons ignore a multiple selection. The keyboard shortcuts only work via the number bar at the top of the keyboard, not via the number pad.

Read

Here you assign reading permissions for the respective document class.

> **Note**
> "Read processing" means here that you can permit a user to already read documents in the processing of a group member.
>
> Via "Read hidden attributes" you can control, if properties marked as hideable are displayed for the user or not (Right rejected).

The following overview describes the rights in greater detail:

Export dependent document: Export of the dependent document.

Export original: Export of the original document.

Read activities: View of the activities.

Read archive: Reading documents in the status “Archive”.

Read attributes: If no reading right for a document exist, then at least the properties can be displayed in d.3 smart explorer, but the document itself can not be visualized.

Read processing: Reading documents in the status "Processing"; also applies if the document is located in your own processing.

Read release: Reading documents in the status Release.

Read blocked: Reading locked documents.

Read verification: Reading documents in the status Verification. This means the unverified as well as the verified documents.

Read hidden attributes: Show hidden properties (as hideable marked property). The right can only be set for all properties marked as hideable in the document class.

Displaying with watermark: **For the visualization of a document a watermark is included and displayed.**

Write

Here you assign writing permissions for the respective document class. You can, for example, also influence the creation of notes and redlinings

Please take care to define suitable combinations of read- and write-permissions. Thus the right **Change hidden attributes** only makes sense if the permission to read these properties is granted.

The following overview describes the rights in greater detail:

Change attributes processing/verification: Changing the properties of documents in the status processing and verification.

Change attributes Release: Changing the properties of documents in the status release and verification. This right is only available, if the parameter **ATTRIB_SUPPORT_MODIFY_RELEASE** was enabled.

Change protected attributes: Right to edit properties even if later changes were disallowed for the property (see Modifiable fields).

Change hidden attributes: Changing the properties which are defined as hidden in the document type.

Update document: Right to change a document (create a new version). The document must be located in the user's own processing the document must be in the status processing of a group of which the user is a member.

Import document: Right to store documents

Change document type: Right to change a document type. This way you can change advanced properties but not the category.

Creating dependant documents: Right to create dependent documents on new import or a document update; this right does not refer to depending signature files.

Create/change notes: Right to create or change notes for a document.

Create/change redlining: Right to create or change redlinings for a document.

Change color marking: Right to create or change the color marking in d.3 smart explorer.

Delete Archive: Deleting documents in the status Archive. This right is only available, if the parameter **ALLOW_DELETE_FROM_RELEASE** was enabled.

Delete Processing: Deleting documents in the status Processing. The document must be located in the user's own processing the document must be in the status processing of a group of which the user is a member.

Delete Release: Deleting documents in the status release. This right is only available, if the parameter **ALLOW_DELETE_FROM_RELEASE** was enabled.

Delete Verification: Deleting documents in the status verification. Additionally, the rights **Status change verification** for the document is required.

Status transfer

Changing the document status can also be controlled using a profile. The fine granulation is obvious.

> **Note**
>
> In order to use a workflow manually, the extended rights concept requires that all seven status transfer possibilities have been set.
>
> The right to withdraw another user's processing is often assigned to key-users or people in charge of a process to allow to react if employees on sick-leave have important documents in their processing.

The following overview describes the rights in greater detail:

Transfer archive: Status transfer of a document into the status “archive”.

Status transfer processing: Status transfer of a document into the status processing.

Status transfer "Withdraw Processing": Withdrawing the processing from another user.

Status transfer release: Releasing a document.

Status transfer verification: Verifying a document, finish verification.

Status transfer verification: Status transfer of a document into verification, verification pending.

Status transfer block: Blocking a document, release version blocked

Links

Documents can be manually linked easily by drag & drop. To create a "controlled" link, define here, what or to what can be linked.

The following overview describes the rights in greater detail:

Create link (superordinate document): Right to select documents of the document class as the target of a link. A link between two documents can only be created using a combination of the rights **Create link (supoerordinate document)** and **Create link (subordinate document)**.

Create link (subordinate document): The right to select documents of this document class as link sources. A link between two documents can only be created using a combination of the rights **Create link (supoerordinate document)** and **Create link (subordinate document)**.

Remove link (superordinate document): Right to remove the link between two documents, where documents of the document class are the target of the link. A link between two documents can only be removed using a combination of the rights **Remove link (superordinate document)** and **Remove link (subordinate document)**.

Remove link (subordiante document): Right to remove the link between two documents, where documents of the document class are the source of the link. A link between two documents can only be removed using a combination of the rights **Remove link (superordinate document)** and **Remove link (subordinate document)**.

To permit the creation of links explicitly, two entries are required:

The right to create a link not only has to be assigned in the child element but also in the parent element.

Activities

About the activities

By the activities in d.3, the user finds out, how the life cycle of a document or a dossier looks like concretely. All user interactions and events as e.g. the import of a new document, a new version, changes of properties, workflows or new comments can be retraced and are listed in a chronological history. You can see, when which activity was done by which user or by the system. Thus, a project manager can see what happened in a certain project dossier lately.

The activity stream of a document or a dossier is displayed via a HTML-interface provided by d.3 server. The user can now see the history of changes to a document or dossier on the tab of the details view. The user can choose between two detail level: **Compact** means that only the important activities are displayed and for **Advanced** all activities are displayed.

An assigned right is required to enable this.

The display of the activities for documents can be controlled with the right **Read activities**. This is not active by default and must be enabled manually in the authorization profile. If the function is desired general, the script **set_activitystream_rights.jpl** to set the right can be used, so that the activities for the document/dossier can be displayed on existing right to read.

Configuration of the HTTP-interface

d.3 configuration

The HTTP interface of d.3 server is enabled by default. To disable this, you can set the parameter **ENABLE_HTTP** to **No** in d.3 config in d.3 admin. Now no new features can be used, if they require this interface. This affects currently:

- The view of activities in each form (document-specific, user-specific, system-wide)
- The welcome page

Configuration of the reverse proxy

d.3 server communicates usually by d.3 gateway with the d.3 applications. If the HTTP interface is used, then d.3 gateway uses additionally d.3 presentation server gateway as reverse proxy. The necessary steps can be found in the hints on the initial configuration and operation at the end of the setup of version 8.0.

Defining the right

By default none user has right to see the activities.

Individual rights assignment

The right to view the activities is called **Read activities**. It can be assigned or revoked in an authorization profile in d.3 admin.

Global rights assignment

Alternatively, the right to view the activities can be assigned to all users, which have the right to read the document. For this effect the external JPL script **set_activitystream_rights.jpl** (you can find it in the d.3 installation directory **\d3\d3server.prg\ext_jpl**) can be used. Then, follow the instructions in the script.

Configuration of the welcome page

On the welcome page a customer-specific website as well as optional two views for activities are displayed. Via the tab **My activities** and **All activities** you can switch between the views.

Enabling the activity view

There are two views for activities on the welcome page.

- Activities of the current user
- Activities in the entire system

For privacy reasons both views are disabled by default. To enable this, you have to do the following settings in d.3 config in d.3 admin.

- Current user: **HOME_ENABLE_OWN_ACTIVITIES = Yes**
- Entire system: **HOME_ENABLE_ALL_ACTIVITIES = Yes**

The parameter **DETAILVIEW_ENABLE_ACTIVITIES** enables the document-specific activities for all users.

Activity view in d.3 smart explorer

Activities lists the most recent activities for the user or for the entire system. This view is enabled by default by the parameter above and can be disabled for e.g. certain groups by the following setting:

d.3 client profile: **d.3 admin > System settings > d.3 client config > d.3 smart explorer > Main Window > Activities (navigation)**

Tracking activity stream of a document as user

Here, all changes to a document are displayed as an activity stream, for dossiers this includes the changes in its content.

The display of the activity stream for documents can be controlled with the right **Read activities** for the document class and can also be disabled for individual groups.

d.3 client profile: **d.3 admin > System settings > d.3 client config > d.3 smart explorer > Main Window > Drop down menu > View > Activities**

Creating a customer-specific website

In the installation directory of d.3 server you can configure a customer-specific website in the folder **html/events/home**. This will be displayed left-aligned on the welcome page. By default a placeholder is configured here. The website must be in the file **welcome.html**. Referenced files can be also configured in this folder.

Authorization profile assignment

This overview shows which authorization profiles are assigned to an user.

Having selected an user or an user group the authorization profiles are displayed in the overview.

Using the checkboxes, the overview can be filtered.

Full name: the full name of the user combined with the technical name

d.3 distribution group: Distribution of documents to all group members

d.3 structure group: no distribution, only used to group the members.

d.3 job profiles: Distribution to one member of the group (mailbox with load balancing)

If an user or a group has not been assigned any authorization profiles, the list remains empty. For detailed information about an authorization profile assigned to the user/group, double-click on an authorization profile in the overview.

An additional window is opened showing all classes of the current authorization profile.

New authorization profile assignment

You can create a new assignment as follows:

1. Click **New**.
2. In the first dialog initially choose all profiles you want to assign.

User assignment

Choose all users, user groups or job profiles for this assignment.

You can differentiate between the following options by means of the preceding symbols:

- Users
- Groups (without mailbox)
- Groups (with mailbox)
- Job profiles (load balancing)

Remove/Edit a profile

1. To remove an assignment of profiles to the user, select these authorization profiles in the overview using the mouse.
2. Afterwards click on **Remove**.

Click on **Edit** to change an existing assignment.

These changes only refer to the currently selected users.

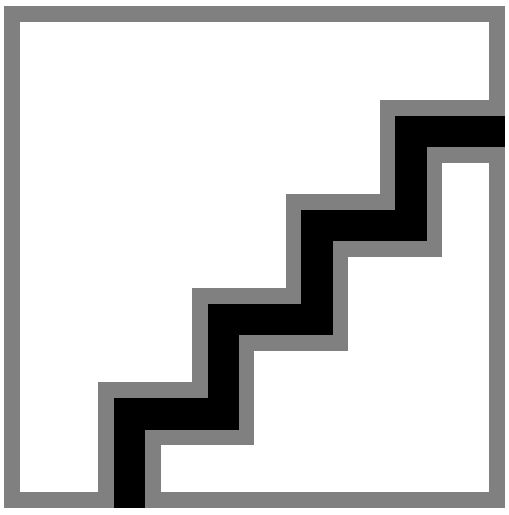
If you want to add additional profiles to multiple users follow the same steps as describer for a new assignment.

If you assign the user "user" to the authorization profile "profile1" and "profile2" this way but this user was already assigned to "profile3", then the user will finally be assigned to three authorization profiles.

It is unfortunately not possible to remove authorization profiles from several users at once.

Complete overview of authorization profile-assignments

The overview shows, which authorization profiles are assigned to which user or which group.



A summary of authorization profiles and the respective user/group is displayed.

Here, the prefixed icon also represents the type of assignment.

User

Mailbox notification without load balancing

Mailbox notification with load balancing

no mailbox notification, grouping only

If you want to save the overview, press the button **Save** or use the keyboard shortcut **CTRL+S**.

User/document class assignment

If an authorization profile is assigned to an user then this is an implicit assignment of rights, as the role itself is linked to one ore more document class, which accordingly controls the access to the documents. In contrast to this you can also explicitly assign classes directly to an user. The effective rights result from the sum of implicit and explicit assignments.

A direct (implicit) assignment can only be performed for one user.

The **Overview of user / class assignments** always relates to one user. When selecting an user in the pull-down list, those document classes are displayed which are directly (explicitly) assigned to this user. In this overview, the names of the individual document classes and the rights are displayed.

In order to change an assignment, do as follows:

1. First, choose a user.
2. Use the option **Edit**.

The mapping dialog is displayed (see the details on Assigning the rights/permissions). You can also edit or remove assignments in this dialog.

You can also create or remove assignments in this dialog.

Temporary document classes

With d.3 version 6.3, the direct delegation of document rights was introduced. This provides differentiated functions to inherit document rights via document classes.

The so called temporary classes assigned by the d.3 server are managed separately from the permanent one defined in d.3 admin.

> **Note**
> The adhoc-delegation of rights only works, if the recipient has got the respective document type in his authorization profile.

Under the **Permanent Document Classes** the “normal" document classes are listed. The list of **Delegated Document Classes** contains those classes generated by the delegation of document rights. These inherited document classes cannot be edited or appended with additional rights by the d.3 administrator.

If the user is "edited", then the temporary document classes are not listed the overview.

If the user has some inherited document classes, then the following dialog is displayed:

The d.3 administrator has the option to remove the document classes assigned by inheritance.

> **Warning**
> You are influencing running processes! Therefore this operation must be confirmed once again!

Sub-administration

Sub-administration allows you to assign administrative rights to other d.3 users. You can configure here, that the respective user only gets certain rights or is only able to see and access to certain options in d.3 admin.

> **Note**
> An administrator can appoint every user to a sub-administrator. A sub-administrator can create no further sub-administrators or change his own rights as sub-administrator.
>
> Furthermore a sub-administrator can not import or export projects/milestones because importing projects/milestones into the productive system without care can lead to inconsistence and this would be a security issue for the productive system.

Adding user as sub-administrator

You have two options to appoint a sub-administrator:

- Create a new user directly with all desired rights.
- Add an existing user to the sub-administrators by assigning the rights in the sub-administrator-dialog.

> **Note**
> Principal rights system:
>
> The right to read effects that the sub-administrator can even see the menu option.
>
> The right to write effects that the sub-administrator can open the module after switching to edit mode. If the right to write is not set, the menu option is grayed out.
>
> The right to write implies the right to read and overwrites it. If the right to write is specified, automatically the right to read is set, too.

To add an already existing user as sub-administrator, do the following:

1. Choose the respective user from the user overview via **Rights > User**.
2. Open the **Properties** of the user.
3. Go to the **advanced properties**.
4. Assign the **Sub Administration access** in the rights section under **User > d.3 > User account** of the rights overview.

   
5. Then, switch from the menu option **Rights** to **Sub-administration**.
6. In the Sub-Administrators Overview dialog, select the desired user, now listed under Sub-administrators.
7. Then right in the rights areas the desired menu option of d.3 admin below **d.3 administration**.
8. Below you can find the rights overview, where you can define the respective right via click in the column Assignment or via the buttons **Assign**, **Revoke**, **Ignore** below.
9. Save the changes with **OK**.

The following illustration shows exemplary that an user has the right to read for the overview of users but can not make any changes.

> **Note**
> How can I create an sub-administrator, who can only change the user's password?
>
> If you crate a sub-administrator and assign the right to read to users, nothing else is needed to change the password in the user dialog.

Below **Rights > Sub-administration** you can find the earlier selected user in the overview of sub-administrators. You can still assign or revoke rights here.

> **Note**
> The sub-administration is also available for groups. The rights for sub-administration can be assigned to all groups as well. The users of the group get the rights because the sub-administration works on basis of the effective user rights. You can see under Rights analysis where the sub-administrator has got the rights from.
>
> LDAP-user can also be sub-administrators.

Rights Analysation

The plugin for the rights analysis is intended to provide you with an overview of the effective rights which a user has for documents and of the document classes and policies.

1. First, choose an user, for which you want to retrieve the rights.
2. Then select the desired type of rights (user or document class) for this user.

Rights type user

In the view for the rights type user, the advanced rights (policies) are displayed in the lower rights section. The tree structure on the left contains a main node and multiple sub-nodes. Here, you choose for which level a policy is to be displayed. Additional information can always be obtained in the statusbar of the window.

Using the main node, you see the effective rights of the user. Accordingly, all rights in the lower rights section are defined as **Yes** or **No**. If the rejection of the right (**No**) results from a direct assignment of **Right denied** or from an inherited **Ignore Right**, cannot be seen on the level of the effective rights. If you, however, select a sub-node wherby user, user group or organizational unit is chosen, then the rights are displayed which are assigned on this level.

Using the selection in the upper right section you can further filter the view.

> **Note**
> Please note that an empty entry on this level means **Right ignored**. Only when determining the effective right in the main node this may become a **No**.

Rights-type document class

If you want to get an overview of authorizations for a user based on the document classes, then select the rights type **Document class**.

The left window shows all document classes assigned to the user while the upper-right window lists the document class-rights as filters for selection.

General rights for the document type

On selection of a folder- or document type you initially get an overview of all document- and folder types of the selected user. The bottom-right section of the window displays the general rights for the selected document type.

> **Note**
> Specialty: The rights for hidden properties (**Read hidden attributes**, **Write hidden attributes**) are displayed with respect to their display in d.3 smart explorer and d.3 import.
>
> I.e., the information is given if these fields are available in these applications or not.

Effective rights on document classes

If you choose a document class under the document type in the tree, then the effective rights for this document class are displayed.

Yes: The right was granted

Denied: The right was denied

No: The right was ignored and the default is **No**.

Allocated rights by user, groups or organizational units

Open the left tree structure with the lowest level(s) and select an entry.

You receive the rights which are assigned via this element.

Here, the icons and text labels have the following meaning:

 Direct assignment of a document class or of an authorization profile to the user.

If it is an direct assignment of a document class or an authorization profile can not be seen in this view. You should therefore always specify descriptive names for document classes and authorization profiles.

 Assignment of the document class via a profile (name up to brackets).

The profile was assigned using the group (name in brackets).

The type of the group (load balancing, no load balancing, group without mailbox) can not be seen directly but does not affect the rights.

Rights for a document

Using a document ID, you can determine the rights of the chosen employee for a document. To do so, enter the actual document ID in the field **Document ID**.

In the displayed result, the document type or dossier type to which the document ID belongs is marked in bold. Additionally, the tree structure highlights the document class via which the employee received the rights. The actual rights with reference to the document class can be viewed and filtered as described above.

The properties of the document can be retrieved via **Properties**.

Effective rights for a document

To display the effective rights for a document found via a document ID, click on the button **Rights**.

The following overview is displayed.

Using the selection of the respective layer in the upper-right window, the filtered rights are displayed.

d.3 organizational structure

After the start of the d.3 admin plugin **d.3 organizational structure** the overview is displayed.

The organizational structure allows to create a hierarchy of the employees of your enterprise.

This structure is the basis for the substitute functionality incl. the workload distribution (see: Substitution rules). This allows the superiors to adjust the checkout-staus for their employees, e.g. in case of sick-leave. Substitutes and heads are defined in the group definition.

In the left-hand window, the created structure is displayed as a tree. The individual levels can be opened as required. With selected node - here "Purchasing" - the bottom right window displays the members of the group.

The upper-right window displays all groups for selection which were not yet assigned in the organizational structure. The prefixed icons easily allow to identify the type of group.

The bottom right window always displays the members of the selected group. Thus you can also select a group in the upper-right window. Members could be users as well as user groups.

Add an organizational unit

To create a new organizational unit, do as follows:

1. Select the node under which the new unit is to be added.
2. Click on the button **Add**.
3. Enter the name of the new organizational unit in the appearing window.

The dialog always displays the parent level so that you can check the hierarchy.

Then, the new level is displayed in the tree structure and you can assign groups.

Assigning the user group(s) of the organizational unit

The assignment of user groups to an organizational unit is done via Drag&Drop. This means, "grab" a group in the right-hand selection with the left mouse-key and drag it to the organizational unit to which this group is to belong.

It helps in assigning the groups that the members of the selected group are displayed in the bottom right window.

You can also modify existing assignments via drag & drop. Simply drag a group into another organizational unit.

Delete mapping

If you want to delete an assignment again, e.g. remove a group from the organizational unit, then you have the following option:

Select the group and press the key **Remove**.

Or:

Select in the context menu option **Remove**.

Or:

Drag the group from the upper-right selection window.

On acceptance of the confirmation prompt, the assignment is deleted and the group is available in the selection again.

> **Note**
> If you delete an organizational unit containing group assignments, then the user groups are automatically assigned the next higher level.

Defining the leadership

The context menu allows you to define a group as the head of the organizational unit.

By doing so all members of this group are head of the organizational unit. You have to assign at least one more group to the organizational unit, so that the members of the leading unit are allowed to change the checkout state of the members of the subordinated group. Then all members of the leading unit are allowed to change the checkout state.

In most cases you will define the head of a group when you create a new user group for workload balancing (see Creating user groups, workload balancing).

Users in organizational units

If you want to find out in which organizational units an employee is assigned, then select a name in the **User** picklist.

Then, all levels containing the employee are underlined.

IDP support

d.ecs identity provider (short IDP) authenticates users via different authentication processes such as Windows authentication (Kerberos) with user name and password against different systems for user administration such as Active Directory (AD) and provides the necessary user information for the use by other apps.

The purpose of d.3 IDP support is that you only have IDP as the leading system for user administration. This saves you from having to manage users on third-party systems such as Microsoft Windows and on the d.3 level twice.

From d.3 version 8.2 on, the IDP support is integrated in d.3 admin. It offers the possibility to couple user groups of IDP with d.3 user groups as well as d.3 authorization profiles and thus to grant users rights to d.3 objects.

System requirements

For IDP support of a d.3 system, d.ecs infrastructure setup must be installed at least version 1.2.2.

In d.ecs http gateway the d.3 system must be registered as app d3.

In the d.3 repository the parameter **ENABLE_HTTP** is enabled by default.

Enabling IDP support

Use the d.3 configuration parameter **AUTH_SYSTEM** to configure the authentication system for the access to the d.3 system. Set the value of the parameter to IDP here. You can find this parameter in d.3 config under **d.ecs infrastructure**.

Additionally you have to set the data of a service user for the access to IDP via the parameter **DECS_SERVICE_USER** and **DECS_SERVICE_USER_PWD**.

Save the configuration and close d.3 admin. Restart all d.3 prozcsses in d.3 process manager so that the new configuration becomes effective. Afterwards, restart d.3 admin.

Once the user control over the IDP link is enabled, the IDP is the leading system against which permissions are checked. If the IDP authentication has failed and a login to IDP is not possible, d.3 checks whether the user is a local user and, if so, performs the login.

In theory, all users can log on to the d.3 system who are defined via IDP and who are read out with the IDP assignments of d.3. However, this is only possible if corresponding authorizations have also been assigned to them via the authorization assignment in the d.3 IDP dialog.

With the first login, a d.3 user is created, which can also be maintained in d.3 admin. Changes to the d.3 user have no effect, because IDP is always the leading system and the authorizations assigned via the d.3 IDP dialog are always read in again. To the extent possible, basic authorizations must be configured in the d.3 IDP dialog.

With each d.3 login, the data is transferred to IDP. The login and password are verified via IDP.

In connection with IDP, users are only known to the d.3 system after they have logged in at least once.

The following additional parameters can be set in this context:

AUTH_SYSTEM_NEW_USER_ONLY_MODE: "New users only" mode for transferring users from external systems.

If this mode is enabled, only new users from external systems are adopted to d.3, but no more changes are made to existing users. This mode can be used, if the users are to be administrated and changed in d.3 admin.

The changes are then not overruled by the IDP interface. The defined properties and rights applied via the d.3 IDP dialog then only affect the initial transfer of an external user to d.3.

Default setting = 0 (disabled):

Any changes in the IDP configuration or to the data in the directory service are adopted in d.3.

AUTH_SYSTEM_SYNC_TIME: Time for synchronization of all user from external systems.

If a time (format: "hh:mm") is specified here, a synchronization of the d.3 user administration with the IDP d.3 connections configured via the IDP administration module will be performed every day at this time.

In this case, new users and changed user settings for existing users are transferred to d.3.

Creating and managing IDP mappings

Open the dialog for creating and managing IDP mappings under **IDP support**.

The mapping of IDP data is divided into the areas **Hints**, **Groups** and **User information**.

Any changes that you make to the data in this dialog box are initially flagged internally. The data is not transferred to the server and does not become valid until you click **Save**.

You can click **Reload** to retrieve the data from the server again. Any flagged information that was not yet saved is lost.

Enable the option **LDAP IDP conversion proposals** to receive proposals for converting the previous LDAP configuration.

Please note:

- This option is available only if you have not yet activated the IDP authentication system.
- When you enable or disable this option, the dialog is reloaded. Any flagged information that was not yet saved is lost.
- The proposals are displayed together with the mappings defined previously in the **IDP mappings** area under **Groups** and **User information**. The proposals are displayed in blue and the existing data is displayed in black.

IDP hints

If you have enabled the option **LDAP IDP conversion proposals**, hints may also be generated while determining the proposals. For instance, you get hints when LDAP settings cannot be mapped directly to IDP settings. You are shown these hints as a list under **Hints**.

IDP group mapping

Under **Groups**, you can configure which IDP groups you want to map to which d.3 user groups or d.3 authorization profiles.

Creating IDP group mappings

You want to create a new IDP mapping. You can assign d.3 objects to multiple IDP groups. You can either assign a d.3 user group or a d.3 authorization profile.

This is how it works

1. Under **IDP groups**, select an entry from the picklist.
2. Under **Object type**, select the mapping type (user group or authorization profile).
3. Under **d.3 objects**, select an entry from the selection list.
4. Click **Add** to confirm the selection.

You have created a new entry under **IDP mappings**. The entry is displayed in green.

Deleting IDP group mappings

You want to delete an existing IDP mapping.

This is how it works

1. Select the IDP mapping to be deleted.

2. Click **Delete**. Note the following during deletion:
   - If the selected entry is a child item in the list, only this IDP mapping is deleted.
   - If the selected entry is a parent item on the main level, all its corresponding child IDP mappings are deleted.

The deleted entries are flagged for deletion internally and displayed in red with strikethrough.

Accepting IDP group mappings

You want to accept an automatically proposed IDP group mapping.

This is how it works

1. Select the relevant entry (displayed in blue) in the **IDP mappings** area.
2. Click on **Accept**. Note the following when accepting proposals:
   - If the selected entry is a child item in the list, only this proposal is accepted.
   - If the selected entry is a parent item on the main level, all its corresponding child proposals are accepted.

The accepted proposals are flagged for acceptance internally and displayed in green.

Rejecting IDP group mappings

You want to reject an automatically proposed IDP group mapping.

This is how it works

1. Select the relevant entry (displayed in blue) in the **IDP mappings** area.
2. Click **Reject.** Note the following when rejecting proposals:
   - If the selected entry is a child item in the list, only this proposal is discarded.
   - If the selected entry is a parent item on the main level, all its corresponding child proposals are discarded.

The rejected proposals are flagged internally and displayed in blue with strikethrough.

IDP user information

Under **User information**, you can choose which user information fields in the user administration (IDP) are mapped to which d.3 user fields.

Creating IDP field mappings

You want to create a new IDP field mapping. If you have defined an LDAP server as a user provider in your user administration (IDP), you can define the user property fields for this LDAP server as d.3 user properties.

Example: If the field **streetAddress** is defined for the street name in your LDAP server, you can assign this property to the d.3 field **Option 1**. In d.3, the field **Option 1** then displays the corresponding street name for the user.

This is how it works

1. Select a d.3 user field in the **IDP mappings** area.
2. Under **Field name**, enter the name of the IDP property to be mapped to the d.3 field. You can determine the name of the IDP property in your LDAP server.
3. Click **Add**.

You have created a new entry under **IDP mappings**. The entry is displayed in green.

Deleting IDP field mappings

You want to delete an existing IDP mapping.

This is how it works

1. Select the IDP mapping under **IDP mappings**.
2. Click **Delete**. Note the following when doing so:
   - If the selected entry is a child item in the list, only this mapping is deleted.
   - If the selected entry is a parent item on the main level, all its corresponding child mappings are deleted.

The deleted entries are flagged for deletion internally and displayed in red with strikethrough.

Accepting IDP field mappings

You want to accept an automatically proposed IDP field mapping.

This is how it works

1. Select the relevant entry (displayed in blue) in the **IDP mappings** area.
2. Click on **Accept.** Note the following when accepting proposals:
   - If the selected entry is a child item in the list, only this proposal is accepted.
   - If the selected entry is a parent item on the main level, all its corresponding child proposals are accepted.

The accepted proposals are flagged for acceptance internally and displayed in green.

Rejecting IDP field mappings

You want to reject an automatically proposed IDP field mapping.

This is how it works

1. Select the relevant entry (displayed in blue) in the **IDP mappings** area.
2. Click **Reject.** Note the following when rejecting proposals:
   - If the selected entry is a child item in the list, only this proposal is discarded.
   - If the selected entry is a parent item on the main level, all its corresponding child proposals are discarded.

The rejected proposals are flagged internally and displayed in blue with strikethrough.

LDAP support

For more information about the LDAP support see the documentation d.3 admin ldap.

d.3 policy manager

d.3 policy manager allows to assign extended rights for users, groups and organizational units. The logfile is divided into three sections:

- User account
- Applications
- Document rights

> **Note**
> If a right was denied once, be it directly for a user or a user group etc., the right for the respective user is always denied. If a right for a user is overall set to **Ignore**, then it is usually denied. If there are exceptions to this rule, then these are listed below.

User-rights or "Policies" are rights or properties which can be asisgned to users as well as to user groups or organizational units. This means that a right for a user can also be derived from his membership in a user group.

Applications

Web access: Users with this right can use the web-search- and import-components. This right must be explicitly assigned when creating a new user.

Upload workflow: This right grants a user the premission to upload workflows to a server. With disabled extended rights, all users have this right. Enabling the extended rights, the user no longer have this right. Accordingly, the right must be explicitly assigned when creating new users.

Release workflow: This right grants a user the permission to release workflows on a server. With disabled extended rights, all users have this right. Enabling the extended rights, the user no longer have this right. Accordingly, the right must be explicitly assigned when creating new users.

Disable workflow: This right grants a user the permission to release workflows on a server. With disabled extended rights, all users have this right. Enabling the extended rights, the user no longer have this right. Accordingly, the right must be explicitly assigned when creating new users.

Workflow inspector: This right grants the user the permission to log in to the d.3 flow inspector and to monitor and administrate workflows there. This includes the permission to remove documents from a selected workflow or to move them within it.

Displaying the audit trail (everything): This right grants the user the permission to view all protocols.

View mailboxes of other users: This determines who is allowed to see other users' mailboxes. Requirement: a configured organizational structure defining the hierarchy. Please also note the configuration of the user groups, where you can define delegation rules for the load balancing.

Access to JPL scripts: Execute user-specific JPL scripts (for internal use, project-related)

Releasing incompletely signed documents: Even if not all signatures exist, the document can still be released.

Changing the e-mail notifications: d.3 users can use the option **E-mail notification** in d.3 smart explorer to specify, whether they want to be informed by e-mails about messages or workflow tasks.

As a d.3 administrator, you need to do the following so that your d.3 users can use the **E-mail notification** function:

- In d.3 admin, you need enable the function to send e-mails.
- The d.3 users must also be authorized to change the settings. If you did not assign the users the permission to change, globally defined values from d.3 admin are displayed. As a result, users can only call up and view the settings, but cannot change anything.

> **Note**
> The rights for workflow functions and e-mail notifications cannot be assigned directly by the LDAP plugin. Instead, they must be assigned to a user group in d.3 policy manager. You can then assign an authorization profile to this group, to which the users are mapped in the LDAP-tree.

User account

Cannot change password: The user to which this right is assigned will not be able to change his password.

Password never expires: If this rights is assigned, then the user's password is valid forever. This right overwrites the d.3 server setting **PASSWORD_EXPIRE_DAYS**.

Interactive login: The user gets the rights to login to a d.3 repository via d.3 login manager.

In the following, the extended rights of the user account are described in greater detail.

> **Note**
> When creating a new user, this right is automatically assigned. Should you receive the error message "This user is not allowed to login“ (error number 5), then please check, if the respective user has been granted the right for interactive login in d.3 admin.

Account is locked: If this property is granted (assigned), then the user can no longer send any API-calls. However, the user is still "active" in the repository and is displayed in d.3 smart explorer in the list of users for the document forwarding etc.

The lock can be caused by too many failed login attempts etc..

Account is disabled: If this property is granted (assigned), then the user can no longer send any API-calls. The user is no longer "active" in the repository and is not displayed in d.3 smart explorer in the list of users for the document forwarding etc. A user can only be disabled, if he dies not have any documents in his processing and in his mailbox.

Account is hidden: If this property is granted (assigned), then the following applies for the user:

- The authentication is performed and a login for the user via d.3 login manager is possible.
- The user has a right for an interactive login.
- The user does not appear in the list of recipients for the document forwarding (Send to d.3 mailbox).
- The user cannot be chosen as an adhoc-deputy.
- The user is not returned via the predefined dynamic datasets.
- For a mailbox dispatch of the user none entries under "Sent" are saved.

Administration access: The user gets the right to login to d.3 admin. This permission allows him to perform all administrative actions.

Service User: These users are created for services which need to connect to d.3, e.g. d.3 abo service or d.3 iTrieve server.

The same applies for the user:

- The authentication is performed but for this user a log in via d.3 login manager is no longer possible.
- The user has no right for interactive login anymore (implicit rejection). **ValidatePasswordForUser** still works.
- The password of this user never exires.
- This user does not appear in the list of recipients for the document forwarding (Send to d.3 mailbox).
- The user is not returned via the predefined dynamic datasets.
- The user cannot be chosen as an adhoc-deputy.

Delegating authentication: This right makes this user is eligible to trust ta user when he is the caller of the user login. This right is designed to avoid the duplicate authentication and configuration of a trust relationship for services that already perform a user authentication themselves and access d.3.

Delegating function calls: If the user calls API functions, these can be executed by another user ID. The user ID is specified via the import parameter **vice_user** for each API function.

Sub-administrator access: The user gets the right to login to d.3 admin. This permission allows the user to perform all administrative actions. For this, the user must get the required sub-administrator access from an administrator.

Document rights

Inheriting rights: Grants a user the permission to actively inherit rights when sending a document to the mailbox of another user, if the recipient does not have the required rights assigned for the document via document classes.

The permission can be granted for the existence in the mailbox or for a specified time.

Additional information can be found in the manual for d.3 smart explorer.

With the d.3 smart explorer plugin it is also possible, to inherit complete dossier structures.

Delete privileged: If the storage system supports “Delete privileged”, this right can be assigned.

Using d.3 inherit rights manager you can get an overview of the inherited documents and revoke granted rights, i.e. the access to selected documents.

d.3 administration

Below **d.3 administration** you can precisely set read- and write-permissions for the selected users to the individual options and functions in d.3 admin.

Assigning policies

Advanced properties via policies can be assigned to users, user groups or organizational units after selecting the right.

Using filter options you can easily reduce the number of recipients.

For the selected entries, rights can be granted whereby it must be made sure that here the default setting (right ignored) is equivalent to a NO, without a explicit assigned YES the right is thus not assigned.

For this effect, use the buttons **Assign**, **Revoke**, **Ignore** or the context menu.

A specified NO cannot be overruled here by a YES, exactly as in the document classes.

Please also consider that you can assign rights to users as well as to user groups or organizational units and that you might be implicitly granting rights.

Thus, thoroughly plan the assignment of policies and document this.

The d.3 administration plugin Rights analysis allows to visualize an overview over the actually defined rights. However, this does not substitute for a thorough planning!

d.3 inherit rights manager

The d.3 inherit rights manager allows you to get an overview of the inherited documents. The plugin displays who actively inherited the document to whom. Additionally it allows to revoke the inherited documents, so that the access to the document or folder is no longer possible. The entry point is located under the d.3 components. Select d.3 inherit rights manager.

Initially all documents are listed. You can also filter for a delegator or a recipient of rights or both using the drop-down lists.

In the overview you can get the following information:

- The document (**title** or **document ID**)
- The **Delegator**, the **Recipient of rights**
- And the **Expiration date**, if a timestamp was defined.

Additionally, the **type of inheritance** is presented, i.e. if only **reading permissions** or also **write-permissions** were granted.

If you want to remove assignments, then mark these assignments and click on **Delete**.

d.3 inherit rights manager is also available to users made eligible to inherit rights as a plugin in d.3 smart explorer. However, you can then only search for those documents inherited by this user, i.e. the delegator cannot be selected.

The prefixed icon easily shows, if the inheritance refers to a dossier or to a document. Since it is also possible to inherit the subordinate level when inheriting dossiers, this inheritance option is displayed with the checkbox **recursive**.

If you want to display the document ID instead of the caption-field in the overview, then enable the checkbox on the bottom left.

d.3 restriction set manager

In the following you will learn what you can do with d.3 restriction set manager.

Restriction sets common

Using the restriction sets, several individual values for a document class can be specified as set.

This feature allows to collect various value combinations and to assign them using the macro @D3SET when defining the class restrictions. The advantage of this procedure is that it reduces the need for hook functions for the control of rights and permissions, which were regularly required, if the restrictions for a property were derived from several values, such as the assignment of a character range (e.g. A-H, etc.) or a date range.

The term "restriction set" or "set" characterizes the following:

- A set defines one or more filter criteria for an property of a document class.
- A set can be assigned to a user, a group, an activity profile, or to an organizational unit.
- A set is defined as a filter for a property field in a document class using a macro (@D3SET (set_name)).
- At runtime, all restriction sets are assigned during the interpretation of the document classes via the set names. The sets allocated to a user using the direct user assignment and using the membership in user groups are combined in a set union. This set union is then used to determine the permission to documents.
- Sets can be sub-administrated (created, edited) by different users. This is implemented through an extension for d.3 smart explorer (plugin: **d3setmgr.dxp**).

Using the previous rights concept (up to d.3 version 6.1) regularly reached certain limits.

The creation of document classes was often tedious or had to be amended by hook functions which could significantly increase the processing time when determining the effective permissions.

> **Note**
> The definition of document classes often used property restriction in the form of
>
> - ranges (cost center 100-200) or
> - an OR-search (cost center 100 or 200 or 300 or ... or ...)
>
> *[de] benötigt.*
>
> This was not possible up to d.3 version 6.1 (except for the ranges in numeric fields) so that you had to create several document classes instead.

Using the restriction sets, the following features were added with d.3 version 6.2:

- Creating more flexible filter criteria for properties in a document class.
- Dynamic compilation of the property filter for a document class with a combination of various sets during runtime.
- Administration of sets by sub-administrators such as heads of department or key users.

Case studies

Below you will find a few examples.

Example 1: Limiting access to document types

The document type “Invoice” contains a field “cost center”. This field can contain the values from "100" to "999". The d.3 permissions should be based on this field.

For this effect, a document class "Invoice access via cost center" is created.

The field "cost center" is defined with the set "cost_center" as a filter using the macro @D3SET (accounting_range).

In the d.3 system, the set "cost_center" is created.

The document class “Invoice access via cost center” is assigned to the user “User1” either directly or via an authorization profile. Furthermore, this user is a member of the group "Department 1".

As long as this set does not contain any filter criteria, the "user1" can access normal to the documents of “Invoice” regarding to his rights (group membership, assignment of document classes/authorization profiles).

In the first step, you assign the set "cost_center" to the group "Department 1" and defines the filter criteria "100;200-400". The members of the group "Department 1" from now on only have access to invoices in the accounting ranges 100 and 200 to 400.

If a user who is a member of the group "Department 1" now accesses the document type "Invoice" he will only see the documents where the property field "cost center" contains a value within 100 or 200 to 400.

In the next step, the set "cost_center" is directly assigned to the user "user 1". The filter criteria "! 300;500“ are defined. You have thus defined a second selection of values to the set.

> **Note**
> The "!" must (NOT) be defined in d.3 config under Retrieval_behavior.

If the user "user1" now wants to access to documents of the document type "invoice", the set union is created using the document class "Invoice access via cost center" across all sets with the name "cost center”.

The user "user1" thus gains access only to invoices with the cost center 100, 200-400 and 500 but not to the cost center 300.

Example 2: Managing rights using sets

You can see in example 1 that the authorizations for users can be directly granted or denied using assigned to restriction sets.

With restriction sets, you can let eligible users assign and create such authorizations via the d.3 smart explorer.

This could be implemented as follows:

You define a document class "Management of the set cost center" based on the document type in which the sets are stored. Among other things, this document type contains a property for the name of the set and the object (e.g. user name, group name) to which this set is assigned. You can use the name of the set such as "cost center" as a filter for this document class. For the object, you can either specify a user name or another set with @D3SET(...), which contains several user names.

You can thus apply the permission to create or change sets which can then be used to grant or deny these permissions to users.

Example 3: Using restriction sets

Another example for using restriction sets can be found in the attachment under Example for restriction sets.

Storage of sets

The restriction sets are stored as documents in the document type $Restrictionset$. Furthermore, this facilitates an easy definition of a rights concept for the editing of sets as document classes and authorization profiles allow to assign rights to these documents.

A list of the sets in the system is maintained in the database whereby an additional language-dependent description for the set is also saved there.

There is no versioning for restriction sets in form of the document files (because there are not any) but only for the properties to be viewed in the property history or in the audit-trail.

Creating sets

1. By selecting the plugin d.3 restriction set manager you open the dialog for creating and managing restriction sets.
2. Via the button **Create** you first create the set by assigning a descriptive name. Optionally provide a description of the set.

   
3. Enter a distinct name as well as an explanatory note:

You can see this created set as a container for values against which the property values are checked later at runtime.

The such created sets can now be used in the Document class definition with the macro @D3SET (example: character range S to Z).

Assigning values to the set

In the right area **Assigned sets** you can create a assignment via the button **Create**.

Assigned: Here, you are defining the filters, i.e. the set of permitted/prohibited values of a listed level. Depending on the assignment level, this restriction set will only be considered in determining the effective rights, if the user is a member of the group or the organizational unit or the filter was directly assigned to him.

- of the **Group** or
- the **organizational unit** or

- the filter was directly assigned to him.

Choose the option **No assignment**, if you want to use the criteria globally as base values.

Filters: Here, you collect all allowed or prohibited values. The special characters (compare chapter d.3 retrieval behavior) are also available for the definition.

You can create a negation always with the internal allocated character "!". A list of all comparative values is determined with the runtime, due to that the positioning of the negated entry is optional. However, a negated value cannot be overwritten by various positive associations.

You can enter the values separated by line feeds or use the defined or-character. A total of 150,000 characters can be entered for your filter criteria.

Only with assigning the class directly to a user or via an authorization profile, the actual rights are defined. The actual right/permission significantly depends on the user group or organizational unit of which a member. In addition to this, the restriction set is directly assigned to the user.

Macros in sets

When creating the filters for sets various macros can be applied which are also used in the document classes:

@D3USER

@D3GROUP

@D3USER_OPTFIELD_XX

@D3USER_LDAPDN

@D3REGEXPR

@D3SET

@D3USER_REALNAME

@D3USER_LONG

Generating and updating sets via the hostimport

With the d.3 version 6.3 of d.3 server, restriction sets can also be created by the hostimport.

Configuration

The parameter **D3SET_IMPORT_PATH** defines a directory for the hostimport process, from which set definition files are read. This parameter can be specified in d.3 config in d.3 admin. The file extension for such filter files is automatically defined as TXT.

Provisioning the import-files

As with the ordinary hostimport procedure, this also requires a JPL file and a document file. The document file gets the file extension TXT (see above).

The following parameters must be defined in the JPL file:

set_name

contains the short name of the restriction set and is a mandatory parameter

set_object

contains the object to which the set filter is to be assigned. You can specify the following:

- a user name,
- a group name,
- an activity profile name or
- an organizational unit name

The parameter can be set to a NULL string or left empty. In this case, a set without direct object assignment is created or modified.

The TXT-file contains the filter criteria either delimited with a ";" or are specified in separate lines.

> **Note**
> Existing restriction set-filters are overwritten by the hostimport. Thus, you must always specify the complete filter in the TXT file.
>
> Restriction set which do **not** yet exist are not created. The parameter **set_name** must therefore always contain the name of the existing set.

Administration of sets by other users

The created restriction sets can be administrated by eligible users, i.e. values can be appended or deleted/corrected. This could be a job for a head of department or area manager who want to adjust the short-term or long-term distribution of tasks in his group. A restriction set can generally also be created empty and then be filled accordingly by the person in charge based on the actual tasks.

For this task, a document class with the respective assignments and rights must be created.

Creating a document class for the administration of sets

Initially, create a new document class $RestrictionSet$ as usual for the system document type.

> **Warning**
> The system- and default classes are only available with disabled checkbox.

The properties of the system document type $Restriction Set$ have the following meaning:

- Set_Name: Name of the restriction set(s) (e.g. character range S to Z) which is to be managed externally.
- $Object$: Object to be accessed via this class such as the set(s) of the user cc, the group Purchasing, the organizational unit Dep. Accounting (as described in the examples above). Here, you can also

specify an previously defined restriction set which contains all users, user groups, organizational units etc. for which the sub-administration is to be allowed.

- $Object-Class$: The assignment range of the set.

0: No assignment

1: User

2: Group

4: Organizational unit

> **Note**
> When updating the d.3 repository to the new version the value “3” for an job profile (pre d.3 version 7) is automatically changed to the value “2”.

The data type $Object-Class$ is numeric so that specifying a range is possible (1-3, -2, 2-).

You can directly combine as you know it from "ordinary" document types. Only the entered values of the restrictions are assigned.

In the example above, the access to the restriction set "PLZ 48000 - 48999" (ZIP code 48000...) is defined, however only for the sets of the users.

Examples for additional assignments:

Entering the value "2" alone for the $Object-Class$ and leaving all other fields empty, the eligible user later assigned to this document class can/cannot access all sets assigned to the groups. Keep in mind that only the logical patterns are defined here and that the actual rejected or granted permissions are defined when assigning the document class.

The assignment of the access can also be controlled using a set. For this effect you can create a set such as "My colleagues".

If the access is actually allowed or denied is still only defined in the assigned (authorization profile or document class-assignment) ("Right assigned" / "Right rejected" / "Right ignored").

The set is then selected in the document class-restriction and assigned.

The document class for the set administration is either assigned with the usual profile or via a direct assignment to a user and defined with the respective rights (read, write).

> **Note**
> You can even define a restriction set containing the names of restriction sets. This can then be referenced in the field `$Set-Name$` using the macro @D3SET(xxxxxx). Thus, nesting restriction sets is basically possible under certain conditions.

Set administration using the d.3 smart explorer plugin

For d.3 smart explorer, the DXP **d3setmgr.dxp** (located under **...\d3client.prg\DXP**) must be provided using the client distribution. Using the menu option **Utilities** eligible users can then open the restriction set manager.

With the installed plugin of d.3 smart explorer the eligible users can however only manage the restrictions sets assigned to themselves.

The active user thus can edit the sets for the users and groups and adjust the respective tasks.

However, he cannot create a new set!

Watermark

From version 8.1.0 d.3 server supports the display of documents with an automatically added watermark. The simultaneous use of d.ecs sign and watermark in d.3 is currently not yet supported.

Several watermarks can be defined and document classes can be assigned. This way, the watermarks are rendered based on the ownership of a document to the document classes with the document file for the display of a document.

Note the following when using the extension **hashcheck.dxp**:

If a watermark is added to the document using the watermark functionality in d.3 server, **hashcheck.dxp** detects a change and issues a corresponding message.

Enabling the watermark support

With the d.3 configuration parameter **WATERMARK_SUPPORT** the general support for the display with watermark can be enabled.

You can find this parameter in d.3 config under **System settings**.

By enabling the parameter you can assign the property for users with the help of the document class right "Displaying watermark", so that these users are only allowed to view documents with watermarks.

Note

- An installation of d.ecs pdf extension version 1.4.2 or higher is required.
- The central installation is recommended.
- After a clean installation a restart of the machine is recommended, thus the d.3 server processes start successfully again.
- The display of watermarks only works if the parameter **V8RunningUpdateBackComp** is disabled.
- The display of watermarks is only valid for documents which are stored in PDF format.

Warning

If the parameter **WATERMARK_SUPPORT** has been enabled but d.ecs pdf extension has not been installed or exists in a version smaller than 1.4.2, then the d.3 processes cannot be started anymore!

Displaying documents with watermark

If it is determined for the rights validation for the user that the right "Displaying watermark" and the file to be displayed is a PDF-file (document file is PDF or dependent file is P1), then for the request of the file via the d.3 gateway a routine for the application of watermarks is passed through.

This determines all document classes matching the document. This results in a list of watermarks which are assigned to these document classes.

> **Note**
> A user who has the edit right to the document class can always see in the status Processing the original document without watermark because only this may be changed. These users see also in other status the original without watermark if it is no PDF and the asynchronous generation of the depending P1-file is not finished yet.
>
> As a result for the users who are not allowed to see the document without watermark, the edit right to the document class must be revoked. In this case a view of the document is not possible, if it is no PDF-file and the generation of the depending P1-file is not finished yet.

If no watermark assignment was found, then automatically 2 default watermarks are generated, so that it is always ensured that a watermark is added.

After the application of the watermark the generated file is stored and provided in a temporary section of the d.3 document tree. This file is valid for 15 minutes. So if another download of the document is to be made within 15 minutes after the generation, then the document can be directly provided. Else, the watermark generation is executed again.

The master async deletes automatically the temporary watermark files which have been created in the last 5 days.

Restriction when using watermarks with d.3 mobile

The following restrictions must be considered when using documents with watermarks in the offline section of d.3 mobile:

1. Offline saved documents with watermark, where a timestamp of the download time is added as watermark will no longer be updated.
   Via the call "**Provide now**" in the d.3 mobile app this timestamp on the document will not be updated because the document will not be downloaded again.
2. For changes on user properties to which a watermark has been added will not be updated as offline provided document.
   Via the call "**Provide now**" in the d.3 mobile app no change of a user property is displayed because the document will not be downloaded again.
3. For a change on the watermark configuration offline provided documents with watermarks will not be updated.
   The change will not be displayed in the offline section. Also via the call "**Provide now**" in the d.3 mobile app has not the effect, that this change will be visible because the document will not be downloaded again.

Workaround in the d.3 mobile app:

To get a current watermark for these documents in the offline section, it is necessary to remove the document from the offline section and to add it again.

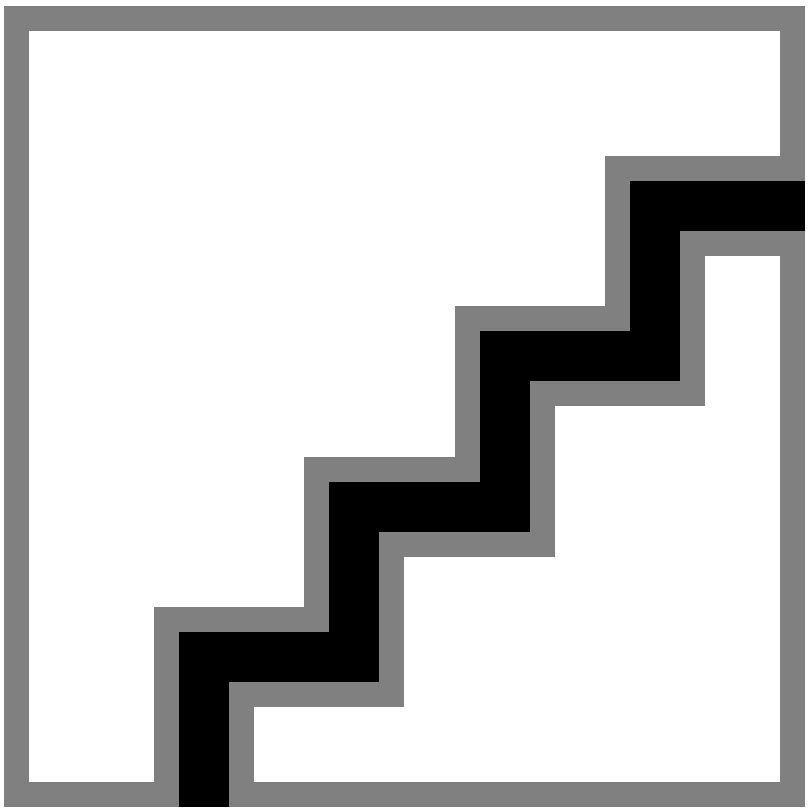
This can be done by this way:

1. Identify the document/dossier in the offline section and e.g. notice the document ID.
2. In the offline section for the document/dossier open the menu option "**Remove from device**".
3. Retrieve the document/dossier via the search for the document ID and provide it offline again.

> **Warning**
> Prerequisite is the use of the software versions d.3 mobile 1.6.0 and d.3 document render service 1.3.0.

Creating and managing watermarks

Open the dialog for the creation and the management of watermarks by selecting the plugin **Watermark**.



Creating a watermark

To create a new watermark use the button **New** in the overview.

General settings

Via the category **General** basic settings of the watermark are defined. This category opens automatically, if **New** was pressed.

Please enter a **Name** for the watermark. This information has only administrative character and is not displayed outside of d.3 admin. The name can have a maximum of 50 characters.

In the **Text** field the components of the text for the display of the watermark can be defined.

This supports the open source template language "Liquid". More on this can be found on the pages https://shopify.github.io/liquid/ or https://github.com/shopify/liquid/wiki/Liquid-for-Designers. Properties of the d.3 document or d.3 user can be included in the text. These properties must be entered within the character string {{...}}.

Possible properties for a document are:

Property	Description	Syntax
doc_id	Document ID	{{doc.doc_id}}
number	Document number	{{doc.number}}
current_status	current document status	{{doc.current_status}}
file_id_current_vers	current file version of the document status	{{doc.file_id_current_vers}}

Property	Description	Syntax
current_version_id	current version number of the document	{{doc.current_version_id}}
editor	Editor of the document	{{doc.editor}}
field[...]	Property fields 1-59 and 70-89	{{doc.field[15]}}
field[6.][...]	Multi-value property fields 60-69	{{doc.field[60][15]}}

Possible properties for a user are:

Property	Description	Syntax
id	User ID in d.3.	{{user.id}}
long_name	Login name in d.3	{{user.long_name}}
name	full name of the user	{{user.name}}
plant	Department	{{user.plant}}
department	Organization	{{user.department}}
opt_field[...]	optional field of the user	{{user.opt_field[1]}}

Examples:

{{doc.field[1]}} displays the value of the d.3 document from the property field at the database position 1.

{{user.name}} displays the full name of the d.3 user.

By enabling the option **Fallback-Watermark** it allows you to configure that this watermark is displayed then, if no other watermark could be determined for the document.

By enabling the option **Watermark always on top** this watermark is always rendered and displayed, also if other watermarks have been added to the document.

Text options

In the category **Text options** you can make settings for the watermark text:

The following options are available:

Color: Use this to set the color for the watermark. The following colors can be used:

- 0: Black
- 1: Red
- 2: Green
- 3: Blue

Opacity: Use this to set the opacity or the transparency for the watermark. The value is specified in percent. The higher the value, the more intense the watermark is displayed. The value 0 results in a transparent and the value 100 results in an opaque representation.

Scaling: Use this to control the watermark's size in percent related to the page. The value 100 extends the watermark across the total width and length of the document.

Alignment

In the category **Alignment** you position the text of the watermark.

With **Position** you specify the horizontal and vertical position of the text. There are 9 anchor points available.

For the horizontal position the following rules apply:

- Left: The text starts at the left margin of the document.
- Center: The center of the text is in the middle of the document.
- Right: The text ends up at the right margin.

For the vertical position the following rules apply:

- Top The text starts or ends up at the top margin of the document (depending on the configured rotation).
- Center: The center of the text is in the middle of the document.
- Bottom: The text starts or ends up at the bottom margin of the document (depending on the configured rotation).

In the section **Offset** the position of the watermark is defined as percentage of the page width and page height.

Horizontal: This parameter sets the horizontal offset for the position of the watermark as a percentage of the page width.

A negative value moves the position to the left and a positive value moves it to the right.

> **Note**
> Negative values may move the text or parts of the text beyond the left document page, while positive values may move it beyond the right document page!

Vertical: This parameter sets the vertical offset for the position of the watermark as a percentage of the page height.

A negative value moves the position to the top and a positive value moves it to the bottom.

> **Note**
> Negative values may move the text or parts of the text beyond the top of the document page, while positive values may move it beyond the bottom of the document page!

Use **Rotation** to control the rotation of the watermark counter-clockwise. The value is specified in degrees. The value 0 shows the watermark horizontally, while the value 90 shows it vertically.

Document classes

In the category **Document classes** you define for which document class the watermark shall be displayed. More on this can be found in the chapter Displaying documents with watermark.

The overview displays the selected document classes.

Via **Add** new document classes can be selected or existing ones can be removed. The following window appears:

Via **Remove** document classes can be directly deselected. A multiple selection is possible here.

Apply changes or reset

If changes were made, then the buttons Apply and Reset are automatically released.

With Apply the changes are saved but not yet sent to the d.3 server. In the overview the name of the watermark appears in italic and bold, if an existing watermark has been changed and in **bold**, if a watermark has been newly created.

With **Reset** the changes are made undone and fall back to the state of the start of the plugin or to the state of the last save of changes.

Save changes

To save all changes on the watermark permanently, use the button Save.

Changing a watermark

To change an existing watermark, select it in the overview.

With **Search for** you can search for specific watermarks by the name.

This shrunks the list via full-text search case insensitive to the results.

Select a watermark and automatically the category **Text options** opens.

The further procedure is the same as described in the chapter Creating a watermark.

Deleting a watermark

To delete an existing watermark, select it in the overview and use the button **Remove**.

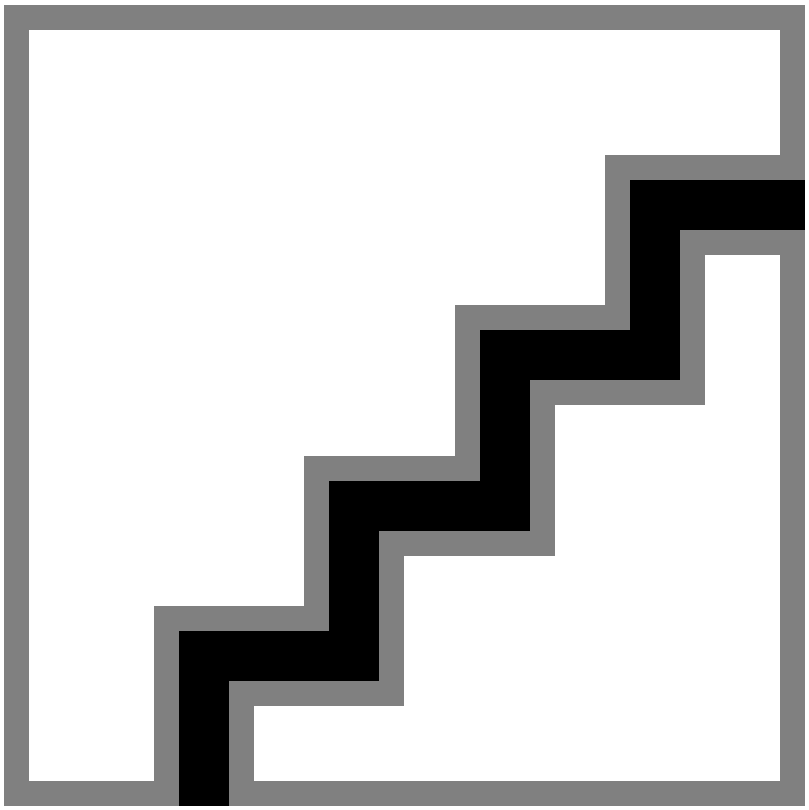
Afterwards, the watermark is marked for the deletion and appears crossed out.

Only, if you use the **Save** button, then the watermark is permanently deleted.

As long as you have not yet pressed the button **Save**, the delete process can be revoked via **Reset**. A watermark which did not exist at the start point will be removed from the list and will no longer be available.

Preview

For a better validation of the settings to watermarks, you can use the preview function of the plugin.



This is located in the upper right section of the plugin and can be used via the button **Preview**. The preview can only be used, if you have selected a watermark in the overview.

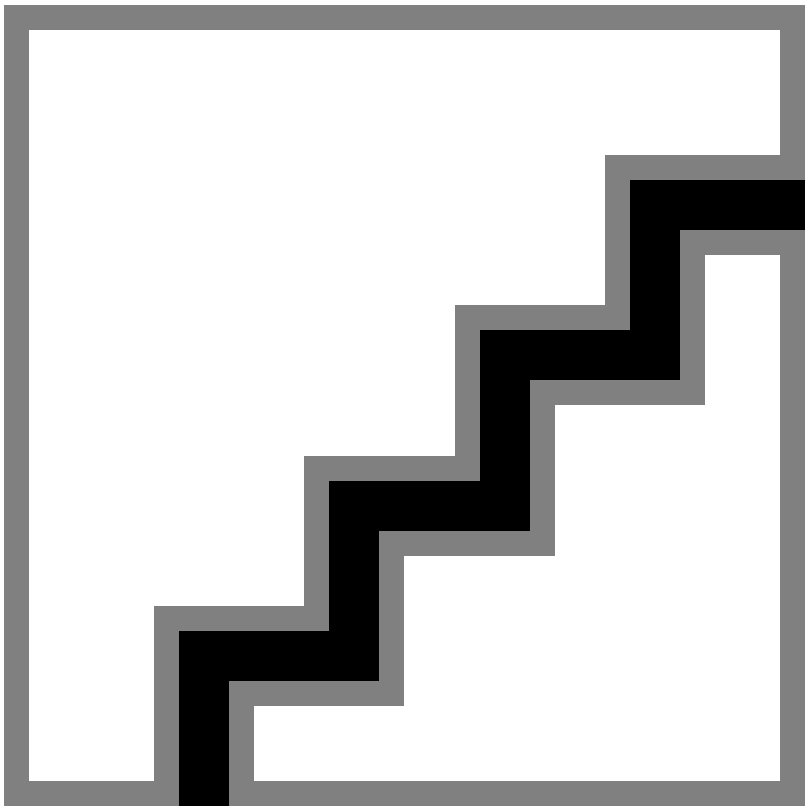
The preview can only be executed for documents existing in the d.3 system. So, initially choose PDF-documents by using the document ID from the d.3 system on which you want to try the watermark.

The d.3 document ID must be entered in the field **Document**.

In the field **User** you enter the ID of a d.3 user in whose name the watermark is to be rendered.

The option field **Simulation** allows you to render also the watermarks already saved in the d.3 system additionally to the selected watermark. This determines the document classes of the specified d.3 document and the assigned watermarks.

By using the button **Preview** a request for generating the preview document is sent to the d.3 server and after successful processing the document is downloaded and displayed in the preview section.



In the preview window you can enlarge the document, scroll through the document or view the document in full-screen mode.

1.4.4. System settings

Directory structure

Documents in the d.3 repository can be stored in different directories on the d.3 server. This chapter describes the differences.

> **Note**
> The directory names are to be viewed from the view of the d.3 server. This means the computer, on which the d.3 server processes (hostimp, d3async and d3server) are active.

If the directory C :\d3\d3.DOK\..... is displayed, then this is the drive C :\ on the d.3 server. Hence, the parameters are only useful, if d.3 admin was started on the d.3 server.

Directories

> **Note**
> With the d.3 version 8.0 the directory structure was changed to minimize the file copy processes. So, a "lifelong" stable storage location is provided and no file operations will be executed by the status transfer. Every 100 document ID a new sub-directory is created. The file structure is: doc_id.file_id+abh. Extension The file_id is not 1:1 equivalent to each document version, that means, gaps between numbers in the file system can occur.

Basically and by default **documents** are located in the directory C:\d3\D3P.dok\docs.

Documents in cache are located in C:\d3\D3P.dok\cached_docs.

Shared files are located in the directory C:\d3\D3P.dok\share.

The **Secondary Storage** is located in C:\d3\D3P.dok\Jukebox.

> **Note**
> The directories can be changed by manual entry in the address field or by a click on the respective icon.

Old directories

The directories below were used in previous versions of d.3 (before 8.0).

Processing

Every user and every group gets an individual processing directory.

For example: You store the file TEST.DOC into the processing of the user USER, then the document will be laid into the directory F:\d3\d3.dok\bearbeit\user of the file server as A0000001.DOC.

Assuming, by storage of the document it has got the document ID F0000001 and your d.3 directory structure is located in F:\d3\d3.dok.

(THIS ASSUMPTION WILL APPLY FOR FURTHER EXAMPLES).

The situation remains the same, if the document was archived into the processing of a group (e.g. F:\d3\d3.dok\bearbeit\group).

Verification

Once the previously stored document is transferred to the status "Verification" with d.3 smart explorer, then the document is moved into the respective directory.

Then, the document is located here: F:\d3\d3.dok\pruefung.

Approval

Once this document is verified and transferred into the status "Release", then the document is moved to the directory F:\D3\D3.DOK\PUBLIC\A000.

As you can see, the document is not directly moved to the **PUBLIC** directory but to the sub-directory **A000**.

This sub-directory structure is automatically generated by the d.3 system. The reason for the use of sub-directories is the increase performance. Under Microsoft Windows operating systems and Novell Netware, significant problems may occur, if one directory hold something like 500,000 files.

There is a further directory for this status under **jukebox**. This is used for the storage on the secondary storage.

Archive

After a further status transfer into the status "Archive", the document is stored in the directory F:\d3\d3.dok\archiv as A0000001.1. There is an additional directory for this status under **jukebox**. This is used for the storage on the secondary storage.

Sec. storage
This directory is required for the communication with the secondary storage system. It contains two sub-directories named **Public** and **Archiv**.

If a document in the status “Release” is to be written to the secondary storage, then this document is copied into the sub-directory **Public**. In our example, this would be F:\d3\d3X.dok\jukebox\public.

For the status "Archive" the directory would accordingly be F:\d3\d3X.dok\jukebox\archiv.

Requests for the retrieval of documents from secondary storage are directly written to the directory **jukebox**.

Notes
This directory contains the note files.

If you added a note to the document in our example, this would be stored in the directory F:\d3\d3.dok\notizen under the file name A0000001.NOT.

Workflow
This sub-directory holds the definitions for the rule-based workflows. It contains a sub-directory named **Protokoll**. In this directory, a workflow-log is stored for every document passing through a workflow.

OCR
This sub-directory contains the OCR files.

Data backup
You can save the actual master data by an automatically backup of the d.3 repository. The so created backup-files are stored in the d.3 document tree in the directory **\D3P.dok\config\d3client\d3admin\mdbackup**. With these files you can either reset the system to a previous state or build up an entirely new system.

The backup can be configured for a daily (at a certain time), weekly (always on monday) or monthly (on a certain day) interval. E-Mails can inform you about the backup job for each time or only on errors.

It is also possible to save the current state by an explicit export. The backup file is stored in the d.3 document tree, too.

> **Note**
> The backup file contains all data which are collected on recording the projects and milestones. Additionally the user data including all memberships in groups and assigning rights are saved. With this data you can reset the system to a previous state or build up an entire new system.

If the backup shall be executed automatically, do as follows:

1. First, enable the option **Automatic backup of all master and status data from the d.3 server**.
2. Specify, how often (**daily**, **weekly**, **monthly**) and at which time (**start time**) the backup shall be executed.

An automatically data backup as a snapshot is performed for every import-transaction of a transport file. This also creates a new milestone in the snapshot project and the current data will be stored there.

You have the opportunity to get informed about the backup by e-mail - always or only on errors.

1. Enable the **E-mail notification**.
2. Choose between **Always notify** or **Notify on error**.
3. Enter your **E-mail address** in the respective field.
4. Click on **Apply**.

Click on the button **Export** to export all master data and status data.

d.3 client config

The d.3 client config module is a plugin for d.3 admin, which is automatically configured with an installation of the d.3 applications (from version 5.5). Additional information can be found in the d.3 admin client config manual.

d.3 config

Every d.3 repository has five configuration files which are usually located under the d.3 server-directory (**d3server.prg**) in a sub-directory named like the short name of the d.3 repository (e.g. **d3server.prg\ \D3FU**, whereby "FU" is the ID of the d.3 repository). This directory contains the files

- **d3admin.ini**
- **d3fixed.ini**
- **d3config.ini**
- **d3addon.ini**
- **aktplan.ini** (obsolete)

With d.3 version 7 the dossier generation definition is administrated in the database. The **aktplan.ini** will be implemented by the update script once only.

The configuration files **d3fixed.ini**, **d3config.ini** and **d3addon.ini** are consecutively loaded on starting the d.3 processes (d3server, hostimp, d3async) in this order. In this process, the settings occurring in several configuration files are overwritten by the setting in the file read last (this is always the **d3addon.ini** last loaded from the d.3 server).

To customize the d.3 repository to your requirements, you can make changes to the configuration files. The file **d3config.ini** can be administrated in d.3 admin. Special settings can be applied in the d3addon.ini.

After the repository installation it always contains the path to the search template-files and the profiles configurable via the d.3 admin plugin d.3 client config.

Example: For the newly created repository "FU" the entry looks like this on Microsoft Windows:

```
CONFIG_PATH_D3CLIENT = „C\:\\d3\D3FU.dok\\config\\d3client“
```

For a Linux system, the following entry would be valid:

```
CONFIG_PATH_D3CLIENT = „/opt/d3/D3FU.dok/config/d3client“
```

> **Warning**
> Never change the file **d3fixed.ini**. This can lead to errors in your d.3 repository.
>
> All changes must be applied in the files **d3config.ini** or **d3addon.ini**.

The **d3fixed.ini** contains all parameters used to configure the d.3 repository or used to create a repository with default settings. If you want to change any parameter you can find information about its usage in the file **d3fixed.ini**.

> **Note**
> Once you changed the **d3addon.ini**, you must restart the d.3 server processes in d.3 process manager for the changes to take effect.

Configuration dialog

The parameters listed and explained below can be easily read and modified with the **d.3 config** plugin of d.3 admin under **System settings**.

The parameters are combined to logical groups. For each parameter, a description, the default value and the currently set value are displayed.

1. Initially, choose the logical group/section to which the parameter is assigned.
2. Choose the parameter via its descriptive name (the technical name is usually not directly telling).

Having enabled the respective parameter, you can choose an entry in the table to enter the change dialog of the parameter.

Sometimes, you can select a new value using a picklist or enter the new value manually or a file dialog is opened. If directory paths are expected, you can open a file open dialog.

> **Note**
> The changes to values and parameter take effect after confirmation via **OK**/**Apply** or **Save**.

Configurable parameters

The following list contains the parameters which can be directly modified in the plugin **d.3 config**.

These are saved in the **d3config.ini** in the configuration directory of the repository under **d3server.prg**.

d.3 document server (document files)

Log every document access

LOG_EVERY_DOCUMENT_ACCESS

A log of every document access and the type of the access is created. This protocol can be used for invoicing purposes (accounting). The data is stored in a database table with the name zugriffs_protokoll.

As soon as this parameter is enabled, every visualization (or actually every action) of the document is written to the table zugriffs_protokoll. First, 100 accesses are cached and then written to the database in a batch to minimize the database load.

action	This column contains the action ID of the change. Every changing action (import, property update, delete) has its own ID. A list of the ID's can be found further below.
tstamp_aktion	Date and time of last change
aktion_id	In this column all changes for every document are unique and ongoing numbered. The aktion_id is required for the link (join) with other tables, especially for the table firm_spez_mul_hist.
aktion_user	For each change in this column the d.3 user is saved, who executed the action.
aktion_text	This column contains a plain text for the change action. The user can enter a text for delete actions by himself.

Action ID:

action	aktion_text
INV001	ImportNewVersion
UPD001	Call of UpdateAttributes
DEL001	Text by user

The parameter only applies, if additionally the parameter LOG_CHANGES_IN_DB is enabled.

Default value: No

Values in the column art_zugriff:

1	View/Download
21	ImportNewVersion
22	UpdateAttributes
31	Deleted in dialog
32	Automatically deleted incl. document file
41	Linked
42	Link removed
51	Status transfer
52	Verified
53	Released
61	Delete a redlining

The table can be interpreted with the additional products Qlikview and d.link for Qlikview.

Column aktion_id:

In this column all changes for every document are unique and ongoing numbered.

The aktion_id is required for the link (join) with other tables, especially for the table firm_spez_mul_hist.

Backup of the attribute data in the database

SAVE_CHANGES_IN_DB

When

- changing the property data of a document
- or deleting a document or even
- creating a link

or

- removing a link

the old values are stored in the database.

Default value: Yes

Creating property backups when manually changing document-/dossier links

SAVE_USR_CHANGES_LINK_DOC
When manually creating or deleting a document-/dossier link, the changes are logged in the database (history table).
Default value: Yes
Creating property backups when automatically changing document attributes
SAVE_SRV_CHANGES_LINK_DOC
When automatically creating or deleting a document-/dossier, the changes are logged in the database (history table).
Default value: No
Creating property backups when manually changing document properties
SAVE_USR_CHANGES_ATTRIBUTES
When manually changing document properties, the old property values are logged in the database (history table).
Default value: Yes
Creating property backups when automatically changing document properties
SAVE_SRV_CHANGES_ATTRIBUTES
When automatically changing document properties, the old property values are logged in the database (history table).
Default value: No
Creating property backups when replacing the document files
SAVE_SRV_CHANGES_REPLACE_USEFILE
When replacing a document file of a document (e.g. by TIFF or PDF), the old property values are logged in the database (history table).
Default value: Yes
Creating property backups when importing a new document version
SAVE_USR_CHANGES_NEW_DOC_VERSION
When manually importing a new document version, the old property values are logged in the database (history table).
Default value: Yes
Creating property backups when importing a new document version by hostimp
SAVE_SRV_CHANGES_NEW_DOC_VERSION
When importing a new document version by d.3 hostimp, the old property values are logged in the database (history table).
Default value: No
Backup of job profile-based resubmission-entries
SAVE_DISTRIBUTION_BASED_HOLD_FILE_ENTRIES
Backup of non-job-profile-based resubmission-entries
SAVE_NON_DISTRIBUTION_BASES_HOLD_FILE_ENTRIES
Information on d.3 mailbox-entries which were sent via an activity profile are logged in the database (history table).
Default value: No
Log property data changes
LOG_CHANGES_IN_DB
Changes to the property data of a document, deleting a document or creating or removing a link are logged in the database.
Default value: No

d.3 database server (characteristics)

Database Management System

db_server

The following DBMS are available for selection:

ORAC	Oracle 7.3 or higher
MSQL	Microsoft SQL Server 6.5 or higher
INFO	Informix 9.3 or higher (is no longer supported with version 7.2)
DB 2	DB2 8.1 or higher
DB2-400	iSeries database

d.3 repositories

Database

List of database-names (ODBC-data source names; DSN) available in the login-dialog for the processes d3_server, d3_async, hostimp etc..

d.3 SMTP Support

Enable SMTP support

SMTP_SUPPORT

Apart from the internal d.3 mailbox functionality, d.3 can optionally send a SMTP-mail to the recipient. On activation of this parameter, the required drivers are loaded.

Default value: No

Enable SMTP for the resubmission

SMTP_MAIL_ON_HOLDFILE

When sending a resubmission for a document, the respective user is notified by mail.

Default value: No

Enable SMTP for the workflow resubmission

SMTP_MAIL_ON_WORKFLOW

When sending a resubmission for a document, the respective user is notified by mail.

Default value: No

Attach document reference to mail

SMTP_MAIL_INCLUDE_DOC

When d.3 sents an e-mail, it attaches a reference to the document. New d.3 smart explorer versions can open this d3l-reference file out of the d.3 repository.

Default value: Yes

SMTP gateway

SMTP_GATEWAY

The SMTP gateway forwarding the e-mail.

SMTP subject line

SMTP_SUBJECT

Subject line of the automatically generated e-mail message.

Default value: d.3 new message: subject_text

Send mail to own mailbox

SMTP_MAIL_ON_OWN_HOLDFILE

On activation an e-mail is also sent, if sender and recipient are equal.

Default value: No

Administrative SMTP-user

SMTP_ADM_MAILBOX_USER

This user account is used for the sending of the d.3 e-mails, when they are sent by system users such as d.3 async.

This parameter does not have a prefixed domain.

Administrative SMTP address

SMTP_ADM_MAILBOX_ADDRESS

E-mail address of the administrative SMTP-user. Some SMTP-servers do not allow sending anonymous e-mails. This could be the case, if d.3 system users send e-mails from within d.3. For these cases, you can specify an e-mail-address for the system users here.

SMTP_MAIL_WITH_URL_ON_HOLDFILE

When resubmitting a document, a link to the document viewer in d.velop documents is displayed in the e-mail.

SMTP_MAIL_WITH_URL_ON_WORKFLOW

In case of a workflow notification, a link to the document processing in d.velop documents is displayed in the e-mail.

d.3 retrieval behaviour

Warning

Note that changes to the retrieval behavior do not only affect the search for documents but also the functions used in document classes.

Before you change anything, please check, if you are using the mechanism to be changed in your document classes and adjust the document class restrictions if necessary.

Use none letters for the customization.

SQL wildcard substitute for %

WILDCARD_JOKER_1

The character '%', which stands for an unknown number of characters in a SQL query, is replaced by this character.

Default value: *

SQL wildcard substitute for _

WILDCARD_JOKER_2

The character '_', which stands for exactly one character in a SQL query, is replaced by this character.

Default value: ?

Delimiter for multiple search

SEPARATE_OR_SEARCH_CHAR

To search for multiple property values within one property field, you can define a key character using as delimiter here.

Example:

SEPARATE_OR_SEARCH_CHAR = "|"

Entry in the d.3 smart explorer: value1|value2

-> Search performed for "value1" or "value2"

Note

Disallowed characters are: Minus- and plus character, colon, slash and comma ('+', '-', ':', '/', ',').

Character for exclusive search

SEPARATE_NOT_SEARCH_CHAR

To exclude property values for a search in fields1 to 49 you can define a character here which precedes a value to be excluded.

Exclusive search via character **SEPARATE_NOT_SEARCH_CHAR** enabled for the following system property fields:

Owner (besitzer)

Document number (zeich_nr)

Editor (bearbeiter)

File name

File extension (datei_erw)

Comments (text)

Example:

SEPARATE_NOT_SEARCH_CHAR = "!"

Entry in d.3 smart explorer: !value3

-> Search everything except "value3"

> **Note**
> Disallowed characters are: Minus- and plus character, colon, slash and comma ('+', '-', ':', '/', ',').

Search character for empty property fields

SEPARATE_EMPTY_SEARCH_CHAR

To search for property fields without entry, you can configure a value here to represent the empty value.

Example:

SEPARATE_EMPTY_SEARCH_CHAR = '$'

Entry in d.3 smart explorer: $

-> Search for all properties without content

Search character for a range search in alphanumeric fields

SEPARATE_SPAN_SEARCH_CHAR

Defines a character to be used to search for a certain range in alphanumeric property fields (1-49, 60-69). This character can also be used for the definition of document classes.

Example:

SEPARATE_SPAN_SEARCH_CHAR = '~'

A search with "A~Z" finds all terms alphabetically following A and preceding Z, e.g. "Apple" or "Wolf".

A search with "Man~" finds all terms alphabetically equal to or following "Man", e.g. "Man", "Mouse" or "Wolf".

A search with "~mouse" finds all terms alphabetically preceding or equal to "mouse", e.g. "Apple" or "Man".

Character for 'less than' search in alphanumeric fields

SEPARATE_LESS_SEARCH_CHAR

Defines a key character that can be used for a less-than search in alphanumeric property fields (1-49, 60-69). This character can also be used for the definition of document classes.

Example:

SEPARATE_LESS_SEARCH_CHAR = '<'

A search with "<M" finds all terms alphabetically preceding "M", e.g. "Land" not "Mouse".

Character for 'greater than' search in alphanumeric fields

SEPARATE_GREATER_SEARCH_CHAR

Defines a character to be used to search for a greater than search in alphanumeric property fields (1-49, 60-69). This character can also be used for the definition of document classes.

Example:

SEPARATE_GREATER_SEARCH_CHAR = '>'

A search with ">Man" finds all terms alphabetically following "Man", e.g. "Mouse".

Key character for combinated search in multi-value property fields

SEPARATE_PAIR_SEARCH_60X_CHAR

To search for multiple multi-value property where several property values are located in the same row (combinated search), a character can be defined which can be used as key for the property fields to be combined by this search.

Example:

SEPARATE_PAIR_SEARCH_60X_CHAR = '§'

Entry in the d.3 smart explorer:

Feld60: §name1

Field61: §value1

Matches where the fields 60 and 61 contain the values "name1" and "value1" in the same row.

Key character for combined search in multi-value property fields

SEPARATE_AND_SEARCH_60X_CHAR

To search for multi-value property fields containing several identical property values, you can define a character to be used as a key character for the search.

Example:

SEPARATE_AND_SEARCH_60X_CHAR = "#"

Entry in d.3 smart explorer: Field60: name1#name2

Result: Matches containing "name1" as well as "name2" in the rows of field 60.

Search for the document status

SEARCH_STATUS_CURRENT

This parameter controls the search for the document status.

If the parameter is enabled (yes), all documents are returned whose current status is the specified one.

If the parameter is not enabled (no), then all documents are returned, having a version in the specified status.

Default value: Yes

Maximum number of multi-value property fields for one search

SEARCH_MAX_COUNT_60X_FIELD

Specifies the maximum number of multi-value property fields to be used in one search. A search across several multi-value property fields can result in performance issues so that you should avoid it whenever possible. This parameter only applies to the search in d.3 smart explorer.

Default value: 2

Search across several document types

DOC_TYPE_MULTI

Similar to the dummy-document type, you can define document types here for which the search across several document types is performed.

Example:

MUL1: Short name of a virtual document type

ART1, ART2, ... ,ART10: ordinary document types

DOKUART_MULTI[1] = "MUL1,ART1,ART7,ART9"

If you now perform a search via the document type "MUL1", then the document type "MUL1" as well as the document types "ART1", "ART7" and "ART9" are browsed.

> **Note**
> The first document type entered here is the dummy document type with the effect that this document type is not displayed in the import application. The parameter **ENABLE_DUMMY_WRITE_ACCESS** allows to control this behavior.

Short name of the dummy document type

DOCTYPE_SHORT_TEMPLATE

Specify the short name of a document type, which can be used as template for a cross-document type search.

> **Note**
> In new repositories, the server no longer returns dummy document types with write access. This has the effect that these document types are not displayed in d.3 import. The parameter **ENABLE_DUMMY_WRITE_ACCESS** allows to control this behavior.

Write access to dummy document types

ENABLE_DUMMY_WRITE_ACCESS

On enabling this parameter dummy document types is displayed in d.3 import if the user has write access.

Default value: Yes

Maximum documents to scan

SEARCH_FETCH_LIMIT

Here you specify the number of documents to be read maximum at an interactive search.

Background: While searching the rights validation/post-filtering starts. This means that it may be necessary

to examine more documents than to be delivered to the user.

Return of the 60 fields in the search as table

SHOW_60X_AS_TABLE

On enabling this parameter the values of different multi-value property fields are interpreted as a related table. The return always starts with the first line.

d.3 document storage

File extension of the note files

NOTE_FILE_EXT

File extension for the file name of note files.

Default value: NOT

Example: "NOT" => For a document with the document ID "IP004711" a created note file receives the file name "IP004711.NOT"

Increment the variant number on import

AUTO_VAR_NR_INCREMENT

If a document already exists for the specified document number and document type then, on import, the variant number can be automatically incremented to the next free number in order to assign a unique key. However, this only works, if no variant number is explicitly specified. Bookmark every newly stored document for later processing with other procedures. Example: all newly imported documents are to be exported again for distribution to other plants.

Default value: No

Automatic document number

AUTO_DOC_NR_ASSIGN_DIALOG

If no "zeich_nr" is specified on import then the document ID is automatically adopted as the "zeich_nr".

Default value: Yes

Additional server IDs

server_archive_ID_array

Up to 50 additional server IDs can be assigned to the document types.

Depending on these server IDs in the document ID the documents can be saved to different sections (pools) on the secondary storage medium.

Deleting documents from release or archive

ALLOW_DELETE_FROM_RELEASE

This allows the deletion of documents from the status release. Up to 5 additional server IDs can be assigned to the document types. Depending on these server IDs in the document ID the documents can be saved to different repositories on the secondary storage medium.

Default value: No

> **Warning**
> However, the following must be considered: First, the user who is to delete a document from status release in d.3 smart explorer must have the necessary rights. Furthermore, the document is not physically deleted, it is rather only tagged in the database that the document is no longer displayed in the result.

Property change for documents in release

ATTRIB_SUPPORT_MODIFY_RELEASE

If enabled, users with the respective right may change the properties of a document even if it is in the status release.

Default value: No

Reason for change on direct status transfer from processing to release

ALTERATION_TEXT_B_TO_F

If the parameter is enabled, user must specify a reason for the change when performing a status transfer from processing directly to release.

No plausibility check of properties on hostimport

NO_ATTRIBUTE_CHECK_ON_BATCH

For the batch import procedure hostimp, the validity of the properties is not checked for advanced property-definitions (datasets, regular expression, min-max-ranges) and with reference for the definition in the document type (mandatory field, etc.).

Default value: No

Validation of numeric property fields

VALIDATE_NUMERIC_FIELDS

When validating numeric fields with partially non-numeric content, the numeric part was retained and no syntax error was returned (Panther standard behavior). On activation of this parameter, this behavior can be changed. For new repositories this parameter is automatically enabled.

Default value: No

File extensions for dependent documents

view_dep_ext_default

This defines the file extensions, which are supported as dependent document.

In this case, the predefined extension have the following meaning:

or	Original
t1 t2 t3	TIFF
a1 a2	ASCII (text file)
n5	Lotus Notes file (e-mail)
p1	PDF
r1	Redlining
s1	Signature:
01	for change original - TIFF document

File extension for viewing dependent documents

view_dep_ext_type

Defines the exact file type of the supported dependent file extensions by d.3.

These can be used for determining the view application of the dependent documents.

Length of the document ID

DOC_ID_LENGTH

Length of the document ID in number of characters for new documents stored in d.3 .

Possible are 8 or 10 repositories, which have previously worked with 8-digit document ID's can be changed to 10-digit document ID's, if the database model with 10 character length is defined for the document ID.

With a 10-digit document ID and a 1-digit prefix for the repository/d.link 999,999,999 documents can be managed in d.3. (Default: 8)

Changing the file extension

SUPPORT_MODIFY_FILE_EXT

On activation it is possible to change the file extension during the lifetime of a document. For every document version, the file extension is stored separately so that each version can be opened with the original application.

Default value: No

> **Warning**
> The visualization of older document versions with modified file extensions only works with d.3 applications from d.3 version 6.3 or with applications supporting the API extensions of the v.6.3 server.

> **Note**
> The update of documents with another file-extension is since d.3 version 6.3 only possible by drag & drop to the d.3 document or via the hostimport.
>
> An update using **Save As ..** creates a new import.

Truncating the property values

CUT_ATTR_VALUES_TO_DEF_LEN

If enabled, the property values are truncated to the length defined in the property repository on import and property update.

During the hostimport the behavior can be controlled with the parameter **CUT_ATTR_VALUES_TO_DEF_LEN** which is directly entered in the JPL file or in the **default.ini** individually.

Default value: No

Number of values per multi-value property field

CUR_60ER_FIELD_NR

This defines the number of values that can be interpreted per multi-value property field (60-field) for the import and the search. The default value is 100. The maximum allowed value for this parameter is 2000.

If the value of this parameter is reduced and documents exists with more multi-value property fields as specified here, then an error occurs on any change to the document and the action will be cancelled.

> **Warning**
> The parameter should only be increased, if you actually need to manage this number of values in multi-value property fields. A high value for this parameter negatively affects the performance during the import, property change and the search.

d.3 user interface and d.3 behavior

Format of date values in title-fields

CAPTION_DATE_FORMAT

For an alphanumeric collation in title-fields containing date-values, the format of the dates can be configured. The abbreviations are supported:

- y for Year
- m for Month
- d for Day

```
CAPTION_DATE_FORMAT = 'yyyy-mm-dd' -> Datumswert = '2012-12-15'
```

d.3 gateway communication

d.3 internal server ID

d3fc_server_id

With this short name, the d.3 server process registers with d.3 gateway. This short name must therefore be a valid d.3 gateway repository ID (server ID). This short name is by default also the prefix of the file name (document ID) which d.3 assigns to the managed documents.

d.3 gateway host

d3fc_server_name

The d.3 server process must register with a d.link service. This requires the name or the IP address of the server hosting d.3 gateway.

d.3 asynchronous processing

Asynchronous processing of receipt-files from secondary storage

ASYNC_INTERPRET_OK_FILES

If this parameter is enabled, then the OK-files from the secondary storage are interpreted by the asynchronous d.3 process. If the parameter is not enabled, a separate process must interpret these files.

Default value: Yes

Directories for asynchronous jobs

ASYNC_JOB_DIR

Directories where the asynchronous jobs are stored.

Defaults for the job-processing

ASYNC_JOB_DEFAULT

Contains, equivalent to the **default.ini** the hostimport, default properties for the jobs.

Job type

ASYNC_JOB_TYPE

Type of the jobs:

MULTILINK: Multiple links

ATTRUPDA: Attribute data-update of existing documents

CUSTOMJOB: Customer-specific async-jobs

Extensions of the property files

ASYNC_JOB_FILE_EXT

File extensions to be used for the job files.

Extensions for the stored property files

ASYNC_JOB_SAVE_EXT

File extensions to be used for the backup of faulty job files. Faulty job files get a different extension so that the job processes do not interpret them again.

Directory for newly generated import jobs

IMPL_FOLDER_DIR

Directory where the asynchronous job processes store newly generated import jobs. The import processes create new folders etc. They are processed by the normal d.3 import (HOSTIMP) process. This directory must also be configured as one of the HOSTIMP import directories.

Extension for import jobs

IMPL_FOLDER_EXT

File extension for the newly generated import jobs.

Default value: fol

Folder for asynchronous import

ASYNC_IMPORT_DIR

Directory where the d.3 server process stores the data for the asynchronous import. These data are then taken over by the HOSTIMP. This directory must also be configured as a hostimport directory including the file extension.

Retention period of successfully processed jobs in the database

D3_JOBS_RUN_OK_LIFE_TIME_DAYS

Number of days that jobs remain in the database, if they have not been completed successfully.

Default value: 0

Retention period of successfully processed jobs in the database

D3_JOBS_RUN_ERROR_LIFE_TIME_DAYS

Number of days that jobs remain in the database, if they have not been completed successfully.

Default value: 31

Delete jobs per cycle

D3_JOBS_DELETE_PER_CALL

Number of old jobs to be deleted by d.3 async with every cycle.

Default value: 500

Recognition of folders

FOLDER_UNIQUE_KEY_NAME

These fields of the database are used for the dossier recognition by d.3 async. This mechanism is used only, if the dossier links are created via `*.A0?`-files (LIN001) and these files contain the parameter **folder_definition** set to "1". For dossier links using the **aktplan.ini**, this parameter has no effect.

Deletion (logic) of documents having exceeded the lifetime

DELETE_DOCS_ELAPSED_AUTO

On activation, the documents having exceeded the time to live are logically deleted if the "event-based deletion" was enabled for this document type.

Default value: No

Deletion of documents (physical) having exceeded the lifetime

DELETE_DOCS_ELAPSED_PHYSICAL

On activation, the documents having exceeded the time to live and which were logically deleted and which remained in the recycle bin for a specified number of days (**RECYCLE_STORAGE_PERIOD**) are also deleted physically.

Default value: No

Delete number of documents per cycle

DELETE_DOC_ELAPSED_PER_CALL

This defines the maximum number of documents to be deleted by the d.3 async in one cycle.

Default value: 50

Deleting documents from secondary storage

DELETE_DOC_ELAPSED_SEC_STORAGE

On activation, for documents having exceeded the time to live which are physically deleted, an according job is written for the d.ecs storage manager.

Default value: No

Retention time of deleted documents in the recycle bin

RECYCLE_STORAGE_PERIOD

This specifies the number of days for which documents remain in the recycle bin after the logical deletion before they are deleted physically.

Default value: 365

Deleting documents in the status processing

DELETE_FILES_FROM_PROCESSING

Specifies, if the document file is to be removed by deleting documents in the status processing. If the parameter is disabled, deleted verification versions can be recovered from the recycle bin.

Deleting documents in the status verification

DELETE_FILES_FROM_VERIFICATION

Specifies, if the document file is to be removed by deleting documents in the status verification. If the parameter is disabled, deleted verification versions can be recovered from the recycle bin.

Use specialized DEL100-async for automatically logical deletion

USE_DEL100_ASYNC

If you need to delete immediately a big amount of documents, this should execute a specialized DEL100-async. Additional information can be found in the manual **d3batch_eng.pdf**.

Automatical deletion of sent mailbox entries

CLEANUP_HOLDFILE_SENT

If this value is set to 0 (default), then sent mailbox entries will bot be removed automatically. Else, this value is for the amount of days after which sent mailbox entries will be deleted after acknowledging by the recipient.

Sec. storage

Addressing secondary storage media

jukebox_daemon

If a document in the d.3 status "Release" or "Archive" is not found on the document-server, then it can also be searched on the secondary storage medium.

Default value: No

Automatic storage on secondary storage

AUTO_DOCS_ON_SEC_STORAGE

Documents transferred to the d.3 states "Release" or "Archive" can optionally be reduplicated to the secondary storage immediately.

Default value: No

Release secondary storage directory

SEC_STORAGE_PUBLIC_DIR

Path to the directory from which the secondary storage retrieves the d.3 documents which are in the status “Release” and are pending for duplication on the secondary storage.

Example: O:\D3\D3.DOK\JUKEBOX\public

Free hard disk space on the secondary storage directory

STORAGE_DIR_MIN_FREE_SPACE

If the storage directory has less free space as this value specifies, then it will be written no longer to the secondary storage. As soon as more free space is available, the storage async will repeat the secondary storage. The free space will be determined every five minutes.

Synchronous storage

EXPORT_TO_STORAGE_SYNCHRON

During the import the files are copied synchronously to the secondary storage and also the outsourcing job is created synchronously. If you need higher performance you can disable this parameter. Then d.3 storage async does the storage asynchronously. d.3 storage async should be monitored to see that it complies with the work.

Secondary storage by d.3 async

ASYNC_CLEANUP_DISK

The d.3 async can free disk space if the secondary storage system is active and the space falls below the threshold defined in the parameter `MIN_FREE_DISK_SPACE`.

Default value: Yes

Lower threshold for free disk space

MIN_FREE_DISK_SPACE

Disk space in megabyte which must be at least available on the d.3 files server (document server). Once the disk space falls below this threshold, then some files are deleted from the file server which were already transferred to secondary storage successfully.

The d.3 async process continues this deletion until the configured number of documents to be deleted per cycle (**DELETE_NR_DOCS_PER_CYCLE**) is reached or until all documents were deleted for which a receipt entry for successful secondary storage exists.

> **Warning**
> In this process, even documents might be deleted where the retention time in the cache was not yet exceeded.

Additionally the parameter **DISK_CACHE_CLEANUP_DEP_DOCS** can influence the cache clean up.

Default value: 500

Delete documents from cache per cycle

DELETE_NR_DOC_PER_CYCLE

Specifies the maximum number of documents to be deleted from the cache per cycle of the d.3 async.

> **Warning**
> If this chosen number is to big, the deletion process can block the asynchronous process of d.3 for a longer time.

Default value: 1000

Deleting dependent documents from cache

DISK_CACHE_CLEANUP_DEP_DOCS

On activation dependent documents are also deleted from the cache.

Default value: Yes

Save document properties

ATTR_TO_XML

All document properties are saved in the d.3 database. If this parameter is enabled, then document properties from status release are stored additionally in the XML files.

These will be stored on enabled jukebox secondary storage to the secondary storage. The secondary storage directory must exist and configured via parameter **D3_VERZ_DOK_ATTRIBUTE**.

Size for the document property file

ATTR_FILE_MAX_KB

If the parameter **ATTR_TO_XML** is enabled, then this is the maximum size for the file containing the document properties to be written to the secondary storage. Such a file can contain the full-text information of multiple documents.

> **Note**
> The size of the XML-file can also be modified repeatedly during runtime. The size is checked after a new property set was attached to the XML file and is then decided, if the file is closed and transferred to the secondary storage. The recommended size is the default of 50 KB.

Default value: 50

Directory for document property files

ATTR_FILE_DIR

Directory where the files with the document properties are to be written. From this place they are retrieved and written to the secondary storage.

Retroactive storage

ASYNC_STORAGE_EXPORT

If this parameter is enabled, then d.3 async will retroactively store documents on the secondary storage which were not yet stored based on the current configuration but have to be stored. Functionality is only executed, if **AUTO_DOCS_ON_SEC_STORAGE** is enabled.

Default value: Yes

Time for the determining of documents pending for secondary storage

ASYNC_STORAGE_EXPORT_TIME

Since determining the documents pending for secondary storage can take some time in large systems, this is only performed once per day and the required information is stored in a separate database table. Here, you can change the time when this should approximately be done (format: “hh:mm”).

Default value: 21:00

Import-procedure HOSTIMP

File extension for action-files

HOSTIMP_ACTION_FILE_EXT

If a hostimport directory is filled with files with various file extensions, then a new hostimport entry (as400_import_*) must be created for every file extension.

Create (as400_import_*).

Using a so called "action-file", an indirect file-import can be created and must only one entry per directory is required. For this effect, an action-file must be created together with the other files for the hostimport. This action file then contains the variable **import_file_ext** with the file extension of the actual document files. The file extension of the action-file is configured in this parameter and must be entered as a file extension for the hostimport in the variable **as400_import_datei_erw**.

Default value: action

> **Note**
> Action files are created by the newer versions of d.link for lotus notes or d.link for microsoft exchange..
>
> Example for an assignment in an action-file:
>
> **import_file_ext = „doc"**

Max number of documents per cycle

HOSTIMP_MAX_NR_DOCS_PER_SESSION

Maximum number of documents which are to be stored in d.3, with every login of the d.3 hostimp. On reaching this number, the d.3 hostimp closes automatically. It can be restarted automatically using the program d.3 process manager.

Default value: 20000

Import directories

HOSTIMP_IMPORT_DIR

Directories containing the documents to be imported.

Example: O:\D3\HOSTIMP

Default values for the properties

HOSTIMP_IMPORT_DEFAULT

Control file specifying the properties equally applying to all documents to be stored from the respective directory (e.g. ....\DEFAULT.INI). These properties will be specified by the document specific properties written in the JPL file. These can be overwritten by the document specific JPL file.

Example: O:\D3\HOSTIMP\DEFAULT.INI

The file extension for the import

HOSTIMP_IMPORT_FILE_EXT

File extension defining the documents to be imported (e.g. TIF, TXT).

Dependent documents for hostimport

HOSTIMP_INCL_DEP_DOCS

Yes: The hostimport browses the import directory for dependent documents with all extensions configured in d.3 as dependent documents (see **view_dep_ext_default**).

No: No dependent documents are imported.

Default value: Yes

Minimum free space on the hard disk

HOSTIMP_MIN_DISKSPACE

The specified number of megabytes must at least be available on the hard disk for the d.3 hostimport processes to operate. Else, the processes stop again.

Default value: 10

JPL file required

HOSTIMP_JPL_MANDATORY

On activation the hostimport expects an property file with the property data for the document in addition to the document file itself.

Default value: Yes

Import directory for restriction sets

D3SET_IMPORT_PATH

Directory containing the files used to create or change the restriction set-assignments.

For a restriction set-assignment, a JPL-file and a TXT-file must be located in the directory.

The JPL file can contain two parameters:

“set_name” represents the name of the restriction set and is a mandatory parameter.

“set_object” specifies, to which object the set filter is to be assigned. You can specify user names, group name, job profiles and organizational units.

The TXT-file contains the set-filters whereby the filters are delimited with a ;-character or a line-break.

Example for .jpl file:

set_name = "Suppliers from A to M"

set_object = "Purchasing"

Ignore additional multi-value property fields

HOSTIMP_IGNORE_ADD_6X_VALUES

Allows to import documents with more than one CUR_60ER_FIELD_NR property values. Here, the additional properties are ignored (Attention: data may get lost).

Default value: No

API logging parameter

> **Warning**
> Log level 9 may only be enabled temporarily, as the debugging mode strongly reduces the server performance.

d.3 async log
D3_ASYNC_LOG_LEVEL 0 = only critical errors 6 = only information 7 = every job being edited is displayed in the log viewer 9 = everything (debuglevel)
d.3 storage async log level
0: Only critical errors 6: Only information 7: Every job being edited is displayed in the log viewer 9: Everything (debug level) Attention: Log level 9 may only be enabled temporarily, as the debugging mode strongly reduces the server performance.
d.3 server log level
D3_SERVER_LOG_LEVEL You can define the range of logging information traced for the individual calls to the d.3 system here: 0 = only critical errors 3 = API-call entry and exit 6 = same as 3, but additional import parameter 7, 8 = same as 6, but additional SQL-commands for search 9 = everything (debuglevel) Default value: 6
d.3 API timeout
D3_API_SELECT_TIMEOUT A search query is aborted having exceeded this time, if the d.3 server process can communicate the cancellation to the SQL-server. Default value: 60
d.3 hostimp log level
D3_HOSTIMP_LOG_LEVEL 6 = only information (default) 7 = Short information on each imported file 9 = everything (debuglevel) Default value: 6
Maximum length for log-rows
MAX_LOG_LINE_LENGTH This value defines the character length of the log entries. Default value: 70
SQL-Runtime-Logging lower limit
MIN_SQL_LOG_RUNTIME_MILLIS All ODBC-operations taking longer than the specified number of an milli-seconds, are logged with the default log level (6) and not with debug level (9).

Crash protocol files (mini-dumps)

MINIDUMP_WITH_FULL_MEMORY

This parameter specifies, if and how the crash protocol files are to be written in case of an error.

Hook functions

Groovy-Hook-directories for customer-specific program adjustments

HOOK_GROOVY_DIRS_CUSTOMER

Specify the directories here, which contain the customer-specific program adjustments implemented by Groovy.

JPL file for customer-specific program adjustments

HOOK_JPL_FILES_CUSTOMER

Name of the .JPL files containing the custom program adjustments (hook functions).

Execute hook functions

HOOK_FUNCTIONS_MAPPING

Entering the JPL hook functions to be executed at the respective entry point (**HOOK_FUNCTIONS_ACTIVATE**).

Groovy functions do not have to be entered here!

On process start, all Groovy files in the configured Groovy hook directories are parsed and the found hook functions are registered automatically. The result of the parsing is returned to the d.3 log and written into a file **log\groovy_hook_functions_registered.txt**.

Activate reload of Groovy-hook-files on change

HOOK_GROOVY_RELOAD_ON_CHANGE

If this parameter is enabled, then saving changes in Groovy-hook-files has the effect that they reloaded immediately. As a result, no d.3 process has to be restarted and the code changes take immediate effect. This parameter can facilitate the developmemnt of a hook function. However, it should remain disabled in a productive environment.

User name for d3fc from hooks

HOOK_DEFAULT_D3FC_USER

Default user for d3fc-calls from hooks.

Password for d3fc from hooks

HOOK_DEFAULT_D3FC_PASSWORD

Default password for d3fc-calls from hooks.

Password identifier for additional DB connections

HOOK_PASSWORD_NAMES

These identifiers can be used as placeholders for passwords in JPL hooks.

The usage is described in "d.3 hook api (jpl)".

Passwords for additional DB connections

HOOK_PASSWORDS

Passwords for replacing the placeholders in JPL hooks.

Parallel use of JPL and Groovy

HOOK_ENTRY_POINT_JPL_AND_GROOVY

If this option is enabled, then you can register JPL hook function and Groovy hook methods for the same hook entry point.

If you have specified the name of a JPL hook function in the "Hook Function" column under "Execute Hook Functions" and an annotation for this entry point exists in a Groovy hook module at the same time, both are registered for this entry point.

[de] eine Annotation für diesen Einsprungspunkt in einem Groovy-Hook-Modul, so werden beide für diesen Einsprungspunkt registriert.

When executed, the JPL hook function is called first, followed immediately by the annotated Groovy hook method(s).

Import

Field number for barcode-status

BARCODE_STATUS_FIELD_NR

Number of the database field holding the status of the barcode.

0: booked

1: scanned

Default value: 88

Remove whitespaces on import

REMOVE_BLANKS_IMPORT

Remove leading spaces in property data on import.

Default value: No

Hash-value creation for single instance check / check for duplicates

GENERATE_DUP_HASH

On activation, hash-values are generated and entered in the database for newly created or modified documents to check for duplicates. Older documents which existed in the repository before the activation of the parameter can be provided with a hash-value by executing the JPL script **fill_empty_hash_values.jpl**.

Default value: No

Check for duplicates

TEST_FOR_DUPLICATES

On activation it is already checked when importing a document, if a document with the same hash-value already exists in the database.

Default value: No

> **Note**
> If such a document already exists, the import is rejected. In case of a rejection by the hostimport, the document files are moved to the sub-directory **duplicates** and additionally a file with the extension .dup is created containing information on the refenece-document. When importing via **ImportDocument** or **ImportNewVersionDocument** the error "10036" is triggered in case of a rejection. Information on the reference-document can then be found in the export-parameter doc_id.

Ignore duplicates in processing and verification

IGNORE_DUPS_IN_B_P

This parameter controls, if the single instance check considers the processing- and verification versions of documents are to be considered because these versions are not audit-proof and can so be deleted at any time. By default these versions are ignored during the check.

Default value: Yes

Reading documents in the status “Archive”

IGNORE_DUPS_IN_A

This parameter controls, if the single instance check considers the archive-versions of documents. By default these versions are ignored during the check.

Default value: No

Ignore duplicates in other document types

IGNORE_DUPS_IN_OTHER_DOCTYPES

This parameter controls, if the single instance check considers the archive-versions of documents. By default these versions are considered during the check. By default documents from other document types are considered during the check.

Default value: No

Workflow

Enabling the workflow

WORKFLOW_SUPPORT

d.3 supports business processes with workflows.

Default value: No

Directory for the storage of the workflow-definitions

WORKFLOW_DEF_DIR

Path for the storage of the workflow-definition in XML-format.

workflow-module-library - server

d3fc_wfl_library_server_name

Hostname of the d.3 server where the workflow-module-library is stored.

> **Note**
> This value must be configured, if the workflow for this repository is enabled but the workflow-module-library is located in another repository.

workflow-module-library - repository ID

d3fc_wfl_library_server_id

Repository ID of the d.3 repository where the workflow-module-library is stored.

> **Note**
> This value must be configured, if the workflow for this repository is enabled but the workflow-module-library is located in another repository.

workflow-module-library - TCP/IP port

d3fc_wfl_library_tcpip_port

TCP/IP port of the d.3 repository where the workflow-module-library is stored.

> **Note**
> This value must be configured, if the workflow for this repository is enabled but the workflow-module-library is located in another repository.

Rendition of the workflow-protocol

WORKFLOW_LOG_TIFF

The workflow-protocols are rendered on completion of the workflow and are attached as dependent TIF `w1` to the document.

Default value: No

Stylesheet for workflow-protocol

WORKFLOW_LOG_STYLESHEET

XSLT-stylesheet used to transform the workflow-protocol and thus defines the layout of the rendered protocol. It must deliver an HTML file.

Default value: **default.xsl**

Write workflow-protocols to secondary storage

WORKFLOW_WRITE_LOG_TO_SEC_STORAGE

On activation, the workflow-protocols are written to the secondary storage disregarding the parameter `SCHREIBE_ABH_DOKS_AUF_JUKEBOX` (in the `d3fixed.ini`).

Default value: Yes

Manual workflow-start for documents

WORKFLOW_START_MANUALLY

On activation: A document can always be manually (e.g. via the d.3 smart explorer) placed in any workflow.

On deactivation: Only documents configured for the manual workflow-start type, can be manually (e.g. via the d.3 smart explorer) placed in this workflow.

Default value: No

Backup of the workflow protocol entries in the database

WORKFLOW_SAVE_STEP_PROTOCOL_DB

If this option is enabled, the protocol entries of all successfully completed workflows are stored in the database (history table).

Default value: Yes

Backup of the workflow variables in the database

WORKFLOW_SAVE_USER_VALUES_DB

If this option is enabled, the used workflow variables of all successfully completed workflows are stored in the database (history table).

Default value: Yes

Statistics

Enabling the statistics

STATISTICS_SUPPORT

If this parameter is set to "Yes" and the SQL-statistics table (statistics_archiv and statistics_run) exist, then d.3 logs statistics for every different file-extension (.TIF, .TXT, ...) and for all document classes regarding the load (number of document IDs, number of physical files, sum total KB). Dependent files were ignored here.

If you want to obtain statistics using the documents and document classes, then enable this parameter. To do so, set the parameter to "1". Afterwards, execute the script **statistik_d3_belegung.jpl**.

> **Warning**
> To do so,first stop the d.3 processes (d3_server, HOSTIMP, d3_async), as well as all other processes which are accessing the database.
>
> Furthermore, you should create a backup of the database.

Start the shortcut **Start | Programs > d.velop > d.3 > d.3 server interface (repository name)** with the dialog for the execution of external JPL scripts. The password of the master user must be entered manually in the login-mask. Then start the external JPL program **statistik_d3_belegung.jpl** via **Programs > External programs** which is located in the sub-directory **d3server.prg\ext_jpl**.

This script browses the entire d.3 repository and initializes the statistics table with the current values. This program must be run once before you can work with the statistics.

> **Warning**
> In large repositories, running the script may take several minutes or longer (up to one hour and more) - depending on the number of stored documents). Please do not abort the script as your d.3 system might otherwise show erroneous behavior.

Default value: No

Statistics interval

STATISTICS_INTERVAL

The statistics for the d.3 repository utilization is in the SQL-tables in certain intervals to facilitate reporting for certain time-ranges later. Allowed values for **BELEGUNG_STATISTIK_INTERVALL** are:

W. weekly

M: monthly

Default value: M

Password

Minimum password length

PASSWORD_MIN_CHARS

Minimum length of the password a user can have.

Default value: 8

Extended syntax rules for passwords

PASSWORD_EXTENDED_SYNTAX

This parameter controls if the passwords must comply with certain rules.

The following syntax-rules apply on activation:

- At least one character from at least two of the following three groups: A-Z, a-z, 0-9.
- At least one special character.
- The password must not contain the user name.

Default value: Yes

> **Note**
> This password setting only affects changes via the Login-Manager.

Time to live of a password

PASSWORD_EXPIRE_DAYS

Time to live of a password in days.

Default value: 10957 (= 30 years)

Enabling the password history

PASSWORD_HISTORY_ENTRIES

If a value > 0 is specified, then the password history is enabled and recent passwords are stored. The parameter value specifies the number.

Default value: 10

> **Note**
> A new password must then differ from those in the history.

Count failed logins

PASSWORD_USER_FAILCHECK

This parameter logs the failed attempts of users to logon. Afterwards, a user is blocked after a specified number of failed login attempts (parameter: **PASSWORD_INPUT_TRIALS**) and optionally unblocked after a certain time automatically (parameter: **PASSWORT_USER_REJECTED**).

Default value: No

Failed login threshold

PASSWORD_INPUT_TRIALS

Number of attempts after which a user is blocked if his password was incorrect. This parameter only applies, if **PASSWORD_USER_FAILCHECK** was enabled.

Default value: 3

Unlock automatically

PASSWORD_USER_REJECTED

Time in minutes after which a blocked user is automatically unlocked. If the parameter is set to "0", then d.3 administrator must unlock the user.

Default value: 60

Change password after unlocking by administrator

PASSWORD_ALTER_ADMINRESET

If this parameter is enabled, then a user must change his password after is was reset by the administrator with the next logon. The parameter does not apply, if LDAP is used. d.3 can only read an LDAP structure such as AD.

Default value: No

> **Note**
> d.3 user passwords can have a maximum of 20 Bytes. This restriction results from the d3fc protocol, where the 20 Bytes are to be set in the header.
>
> For each d.3 API-call user name + password will be transmitted. The password must fit into the call-header.
>
> Longer user passwords can only be supported actually by using session passwords as done by the LDAP-interface with Kerberos authentication.
>
> The usable number of characters can be smaller in unicode mode depending on which (special) characters are used in the password because ASCII characters are coded with multiple bytes.

Bidirectional proxy

Support for the bidirectional d.3 proxy

PROXY_BIDIRECT_SUPPORT

Support for the bidirectional d.3 proxy (with enabled write cache)

> **Warning**
> This parameter must not be simply disabled!
>
> If the bidirectional proxy functionality is used and is disabled, then you must first disable the write cache of all d.3 proxy systems accessing this repository.
>
> Only afterwards, this parameter may be disabled here on the d.3 server. Else, proxy placeholders are still stored but the actual documents are no longer automatically synchronized from the proxy systems to the repository.
>
> Having enabled it, all d.3 server processes must be restarted!

Default value: No

Start time for synchronization

proxy_sync_start_time

Time interval in which the d.3 async process automatically synchronizes the data from the write-cache. If access is required outside this interval, the d.3 gateway fetches the document from the write cache "on demand".

Default value: 18:00

Stop time for synchronization

proxy_sync_stop_time

Time interval in which the d.3 async process automatically synchronizes the data from the write-cache. If access is required outside this interval, the d.3 gateway fetches the document from the write cache "on demand".

Default value: 23:59

Timeout for synchronization

proxy_sync_timeout

Maximum number in seconds that d.3 waits for a document not yet synchronized from the write cache before it sends an "on demand" synch request to d.3 gateway.

Default value: 30

d.ecs rendition service

Support for d.ecs rendition service

DTIFF_SUPPORT

Enabling the optional module d.ecs rendition service (server-based rendition) for d.3.

> **Note**
> The name TIFF in the parameters general substitutes for the rendition of documents and thus refers to the creation of TIFF or PDF documents (PDF only with installed PDF extension).

Default value: No

d.ecs rendition service server

d3fc_dtiff_server_name

Name of the machine where d.ecs rendition service (optional module) is installed. The host name of d.3 gateway must be specified in **DTIFF_LEVEL = 1**.

Via d.ecs rendition service you have the option for the server-based rendition of documents.

Default value: localhost

TCP/IP port for d.ecs rendition service

d3fc_dtiff_server_tcpip_port

TCP/IP port of the machine where d.ecs rendition service is installed. The host name of d.3 gateway must be specified in **DTIFF_LEVEL = 1**.

Timeout for d.ecs rendition service

d3fc_dtiff_server_timeout

Default value: 3404

Maximum time in seconds for which d.3 waits for the (TIFF-)result once it sends the request to the d.ecs rendition service for synchronous processing. This timeout prevents blocked d.3-processes in case of exceptionally long renditions.

After this timeout, rendition-jobs are checked asynchronously by the d.3 async process and the TIFF-document is transferred to d.3.

Default value: 30

Full-text-search (d.3 search)

Support for d.3 search

DSEARCH_SUPPORT

Enabling the use of the optional module d.3 search (external full-text engines) for d.3. For the import, the parameter **KEYWORD_OPTION** must at least be set to "1".

> **Warning**
> The full-text support with a unicode d.3 repository is only granted with d.3 search.

Default value: No

Full-text-option

KEYWORD_OPTION

The following options are available for the full-text management:

0: No words are adopted in the full-text index on import.

1: Only the document properties are transmitted into the full-text index.

2: The document properties as well as the document content (OCR file) are transmitted into the full-text index.

3: The document properties as well as the content of the document file are transmitted into the full-text index.

4: The document properties, the document content as well as the content of the document file are transmitted into the full-text index.

> **Note**
> Prerequisites for the full-text management and full-text search is an installation of the product d.3 search!
>
> For using d.3 search the parameter **DSEARCH_SUPPORT** (see "Support for d.3 search") must be enabled explicitly.

OCR extension

ocr_ext

Extension identifying the d.3 keyword-file.

During the batch-import, a full-text indexing is only executed, if a file with the specified ending exists and the configuration parameter **KEYWORD_OPTION** is set to "2".

The keyword-file should be a pure text-file without special characters. Documents created with word processing-applications such as DOC-files with Microsoft Word are not supported. They must first be converted, i.e. the formatting information must be removed.

d.3 search server

d3fc_dsearch_server_name

Name of the host where d.3 search (optional module) is installed. Using d.3 search external full-text-engines can be integrated.

Default value: localhost

From d.3 search 3.0 on, a d.3 gateway connection is also supported. As name, the server on which d.3 gateway is running must be specified.

TCP/IP port for d.3 search

d3fc_dsearch_server_tcpip_port

TCP/IP-port of the machine where d.3 search is installed.

Default value: 3406

From d.3 search 3.0 on, a d.3 gateway connection is also supported. The default port is then 3400.

Maximum number of full-text matches

FULLTEXT_FETCH_LIMIT

Sets the maximum number of results for the full-text-search.

Timeout for d.3 search

d3fc_dsearch_server_timeout

Maximum time in seconds for which d.3 waits for the result once it sends the request to the d.3 search. This parameter prevents a timeout of the d.3 processes, if the processing by d.3 search takes unexpectedly long.

Default value: 30

d.3 search pre-filter for property fields

DSEARCH_DIRECT_DOCFIELD_SEARCH

Enables the direct search of d.3 search for the document property fields. For a combined full-text- and property field-search, this reduces the result list of d.3 search and accelerates the further filtering by the d.3 server.

Default value: Yes

d.3 search standard corpus

DSEARCH_DEFAULT_CORPUS

Is used if no corpus for the document type was explicitly specified; when empty, this value is derived from the d.3 gateway server ID (**d3[d3fc_server_id]**).

Display of the data types

> **Warning**
> Since d.3 admin Version 7.2.0 the following parameters are no longer displayed and cannot be changed.
>
> Exception: If the parameter was changed in an earlier d.3 admin version and therefore no longer contains the default value, then the parameter is still displayed.
>
> With version 8.0 these parameters are no longer interpreted by the d.3 server!
>
> Check before the update to version 8, if the previously mentioned parameters deviate on your system from the default values. If this is the case, do as follows:
>
> 1. Check the applications communicating with the d.3 server (e.g. webservice, hostim-import directories, ...), to what extent the default values above are not used by the communication and edit these.
> 2. Disable your regarding customer-specific configuration of d.3 server, by setting the default values for the parameters in d.3 admin.

Thousands-Separator for money-fields
FORMAT_7X_THOUSANDS_SEPARATOR
Default value: EMPTY
Decimal-Separator for money-fields
FORMAT_7X_DECIMAL_SEPARATOR
Default value: DOT
Minimum number of Decimal Digits for money-fields
FORMAT_7X_MIN_DECIMAL_DIGITS
Default value: 2
Maximum number of Decimal Digits for money-fields
FORMAT_7X_MAX_DECIMAL_DIGITS
Default value: 2
Thousands-Separator for numeric fields
FORMAT_8X_THOUSANDS_SEPARATOR
Default value: EMPTY
Decimal-Separator for numeric fields
FORMAT_8X_DECIMAL_SEPARATOR
Default value: DOT
Minimum number of Decimal Digits for numeric fields
FORMAT_8X_MIN_DECIMAL_DIGITS
Default value: 0
Maximum number of Decimal Digits for numeric fields
FORMAT_8X_MAX_DECIMAL_DIGITS
Default value: 2

LDAP

Time for synchronization of all d.3-LDAP-users

LDAP_USER_SYNC_TIME

If a time (format: hh:mm) is specified, then a synchronization of the d.3 user management directory service configured in the LDAP-administration module is performed every day at this time. In this case, new users and changed user settings for existing users are transferred to d.3.

Administrative LDAP-user name

LDAP_ADMIN_USER_NAME

If the specified directory service does not allow an anonymous login, then an LDAP-user name can be specified here, which is used during the synchronization to read all user values.

Password for the administrative LDAP-user

LDAP_ADMIN_USER_PASSWORD

If the specified directory service does not allow an anonymous login, then you can enter the password for the LDAP-user to be used during the synchronization here.

Read user data with administrative LDAP-user

LDAP_READ_ACCESS_ADMIN_USER

Enabling the parameter **LDAP_READ_ACCESS_ADMIN_USER** the user data are read by the administrative user (**LDAP_ADMIN_USER_NAME**) from the directory service and not from the login user itself (default).

This can be used, if security policies of the customer-IT only permit reading access to the directory tree for users with elevated rights. In this case, only the authentication (**BIND**) is performed with the user information of the login user. All reading accesses are now performed only by the administrative user.

Validity of the session-password

KERBEROS_SESSION_PASSWORD_DAYS

Validity of the session-password in days. After expiration of the password the system assigns a new password with the next logon.

Default value: 2

Creating session-passwords

LDAP_GENERATE_SESSION_PASSWORD

Default value: Yes

Enabling this parameter, the d.3 server creates session passwords, even if no Kerberos authentication is used.

This way, applications can sign in with the LDAP-user name as well as with the d.3 user name simultaneously for the same user account.

In analogy to the Kerberos-solution, with enabled LDAP-interface, a mixed mode of multiple applications of the same user is supported, disregarding of the fact, if these use the LDAP-user name or the d.3 user name.

However, this requires the support of session passwords by the client-applications. If this is ascertained, then the parameter can be enabled.

> **Note**
> For d.3 applications provided by d.velop this is the case since d.3 version 6.3.1.
>
> If other, possibly customer-specific applications are used accessing d.3, then it must be checked, if these adopt the session-passwords returned by the d.3 server (see manual d.3 API; function `ValidatePasswordForUser` with the parameter `package=Basic`).

The parameter must be enabled in the following situations:

- The LDAP interface is enabled.
- Users are participating in a workflow via d.3 web and simultaneously also use other d.3 applications.

Example: A d.3 user uses d.3 smart explorer and takes part in a d.3 workflow via Microsoft Outlook.

In this case the parameter must be enabled for the simultaneous operation of the user account by d.3 smart explorer and d.3 web flow connector to work.

"new users only" mode for the LDAP-interface

LDAP_NEW_USER_ONLY_MODE

If this mode is enabled, then the LDAP-interface will only adopt new user in d.3 but will no longer apply changes to existing users.

This mode can be used, if the users are to be administrated and changed in d.3 admin.

The changes are then not overruled by the LDAP-interface.

The property and right definitions applied via LDAP then only affect the initial adoption of an LDAP user to d.3.

Default setting = 0 (disabled):

LDAP is the leading system for the d.3 user management.

Any changes in the LDAP configuration or to the data in the directory service are adopted in d.3.

Java/Groovy

Java/Groovy support
JAVA_SUPPORT
Enabling the support for die execution of Java and Groovy code in d.3.
Default value: No
Java Virtual Machine
JAVA_JVM_LIBRARY
Library with desired Java Virtual Machine. This is loaded on start of the d.3 processes to initialize the Java Runtime Environment.
Note For the Oracle Java Runtime Engine (JRE) under Microsoft Windows, the **jvm.dll** and under Linux the **libjvm.so** is used.
JVM options
JAVA_JVM_OPTIONS
Options for the Java Virtual Machine For detailed information see the documentation of the Oracle Java HotSpot VM Options.
Example for setting JVM option values:
If you want to adjust the used memory of the JVM, this can be done via the options "-Xms" and "-Xmx". "-Xms" defines the default volume which is directly allocated. "-Xmx" defines the maximum volume of memory which may be allocated at maximum.
If you want that the default volume is 512 MB and a maximum of 1024 MB are to be allocated, then the configuration would look like the following:
-Xms512m -Xmx1024m
Note A maximum of 1300 MB can be allocated because d.3 server is a 32-bit application.
Java CLASSPATH
JAVA_CLASSPATH
File path to the custom Java classes.
Here you can specify a directory, where the .class files are stored. Alternatively, you can enter the file name of a JAR file.
Several file names separated by a semicolon can be specified.
Java/Groovy API functions
JAVA_API_FUNCTIONS_DIR
Enables the plugin interface for API functions developed in Java or Groovy. Groovy scripts or JAR files, which implements d.3 API functions, are loaded from this directory.
Java Remote Debugging

Java/Groovy support
JAVA_REMOTE_DEBUGGING Start Java Virtual Machine in debug mode. This allows to connect to the d.3 server process using the Remote Java Debugger to debug Groovy hooks executed in it. The port 43400 is used for the communication. As each d.3 process starts its own Java Virtual Machine (JVM), the used ports are incremented. The first process started with **JAVA_REMOTE_DEBUGGING** enabled opens port 43400, the second port 43401 etc. The determined port will be displayed on start of the JVM by message "Java Remote Debugging Port" in the d.3 log. **Note** The JVM is started by d.3 on demand with the first access to Groovy code and is usually not available directly after process start.

Substitution rules

Redistribution of documents after x days
DISTRIBUTION_AFTER_CHECKOUT_DAYS 0: Documents are redistributed immediately after the checkout of a user. > 0: Documents are redistributed by the d.3 async process ... days later. Default value: 0
Ignore read mailbox entries
DOC_DISTRIBUTION_IGNORE_READ If this option is enabled, then only the unread mailbox-entries are considered in the calculation of the workload balancing. Default value: Yes
Document distribution to substitute
SEND_MAIL_TO_SUBSTITUTE If a user checked out and a substitute is defined, then all mailbox-entries can be redistributed to the substitute. When checking in again, all unread entries can be returned to the actual recipient again. To enable this behavior, the parameter **SEND_MAIL_TO_SUBSTITUTE** must be enabled (= "1"). Default value: No
Consider read entries for document distribution
DISTRIBUTE_READ_DOCS On activation the read mailbox entries are considered for the distribution to an adhoc-substitute, too. This parameter affects to the parameters **SEND_MAIL_TO_SUBSTITUTE** and **MAILBOX_SUBST_TYPE**. Note: After the distribution all mailbox entries are marked as unread for the substitute. If a back distribution takes place, all entries are marked as unread, too.
Type of the document distribution to the substitute
MAILBOX_SUBST_TYPE With the parameter **WV_SUBST_TYPE** you can control, if the mailbox entries shall be copied (**COPY**) or moved (**MOVE**). If the entries are to be moved, then the unread entries are returned on checkin. Default value: COPY
Copy to recipient
SEND_DOC_TO_ORIGIN If a document is to be sent to a user who has checked out and entered a substitute, then this is sent to the substitute (usual behavior), with enabled parameter **SEND_DOC_TO_ORIGIN** a copy is sent to the actual recipient. Default value: No
Superior user in optional field of a user

SUPERIOR_IN_USER_OPT_FIELD

1-10: Number (1 to 10) of the optional user fields, where the d.3 user name of the superior is defined.

0: No superior user is defined.

If this value is set to "0" or the optional field with specified number is empty, then the superior is determined using the d.3 organizational structure. Else, this process overrides the d.3 organizational structure.

Default value: 0

Superior on the same level

SUPERIOR_SAME_LEVEL

When determining the superior of a user, all members of his activity profiles are also regraded as superiors. If this parameter is disabled, then the superior must be located at least one level higher in the organizational structure.

Default value: No

Rights validation with substitues at @D3USER

USER_MACRO_USE_SUBST

If you are using the macro @D3USER in a document class, the users defined as substitutes for the caller can also be included in the validation of rights. So the comparison of the character string will be done against the user name of the caller and the user name of the substitutes.

Default value: No

Rights management

Withdrawing the document processing as a superior

REVOKE_PROCESSING_SUPERIOR

Yes: If a document is located in the processing of a user, then only this user and his superior user may edit this document or change its status.

No: Every user with write-permissions for the document may withdraw the document whereby a warning is displayed for the revoking user.

Default value: No

Withdrawing the document as an administrator

REVOKE_PROCESSING_ADMIN

Yes: If a document is located in the processing of a user, then only this user and a d.3 administrator may edit this document or change its status.

No: Every user with write-permissions for the document may withdraw the document whereby a warning is displayed for the revoking user.

Default value: No

Checking rights when sending resubmissions

MAILBOX_CHECK_RECIPIENT_RIGHTS

On activation, it is checked before a Send to Mailbox (only individual recipients), if the recipient is eligible to view the document. If "no", an error message is displayed.

Default value: Yes

Checking rights after a property update

UPD_DOC_CHECK_NEW_CLASS

After an update of properties (UpdateAttributes), it is checked, for the document, if the user still has write-permissions to this document.

Default value: No

Additional rights validation while accessing files

EXTRA_RIGHT_CHECK_ON_FILE_ACCESS

By default, an access key is generated while searching for documents, which allows the d.3 application to display and download the physical files of the document for a limited period of time. If this parameter is enabled, an additional rights validation is synchronously performed while accessing the file. Enabling this parameter results in higher system load.

Automatic blocking on inactivity

BLOCK_USER_AFTER_DAYS
This specifies the number of days which may pass since the last logon of a user, before it is automatically blocked by the system.
Default value: 0 (no automatic blocking)
Automatic deactivation on inactivity
DISABLE_USER_AFTER_DAYS
This specifies the number of days which may pass since the last logon of a user, before it is automatically disabled by the system.
Default value: 0 (no automatic disabling)
Automatic removal on inactivity
REMOVE_USER_AFTER_DAYS
This specifies the number of days which may pass since the last logon of a user, before it is automatically removed by the system.
Default value: 0 (no automatic removal)

Other parameters

Color code for repository
ARCHIVE_COLOR_CODE
Defines the color for the repository icon.
0 (default): Red
1: Green
2: Blue
3: Purple
Sending of UDP packets for object monitor
UDP_SUPPORT
This parameter must be set to "Yes" for the d.3 server, d.3 hostimp and d.3 async to send regular UDP-packages to the monitoring service.
Default value: Yes
Abo service support (subscription service)
ABOSERVICE_SUPPORT
When enabling this parameter, the d.3 abo service (subscription service) process for the d.3 repository can be started and the context menu option **Subscribe** in the d.3 smart explorer is displayed. For users with an assigned e-mail address, the option is then enabled.
Default value: No
Default character-encoding
DEFAULT_CODEPAGE
This value should be set, if all previously stored document properties share the same character-encoding. If the repository contains documents with properties in various different character-encodings, then the value should remain set to "0". Otherwise, the clients might receive an incorrect character-encoding for existing documents.
Default value: 0
Translating datasets in search results
TRANSLATE_SEARCH_RESULTS
If this option is disabled, then the translation for static datasets are only displayed in the input forms (for the search and property change). If this parameter is enabled, then the current properties of a document are also displayed as translated.
Default: Yes
Syntax of regular expressions

REGEX_SYNTAX

If this option is disabled, then the translation for static datasets are only displayed in the input forms (for the search and property change). If this parameter is enabled, then the current properties of a document are also displayed as translated.

Default: Yes

By default, d.3 employs the 'POSIX Basic Regular Expression Syntax'. To be able to create more complex regular expressions, the 'POSIX Extended Regular Expression Syntax' or the 'PERL/JavaScript Regular Expression Syntax' can be used. These types feature a greater function set but differ from the 'basic syntax' in central points.

> **Warning**
> It is urgently recommended to check existing regular expressions in hooks, plausibility-checks of advanced property fields and document-class-filter, if they are compliant with the new syntax.

The 'POSIX Basic syntax' is basically compatible with the 'JPL syntax' but has more functions and is unicode-enabled. Please only use the 'JPL syntax', if you need to be 100% backward-compatible and you do not need the correct handling of unicode characters.

> **Note**
> You get the benefit of the "POSIX Extended Regular Expressions Syntax" using or-expressions.

Example: The regular expression "d.3 is (good|brilliant)!" allows the following examples: "d.3 is good!" or "d.3 is brilliant!".

Default value: POSIX Basic Regular Expression Syntax.

Interpret upper-/lower-case in regular expressions

REGEX_CASE

This specifies, if the interpretation of regular expression is to be case-sensitive. This setting has no effect, if the "JPL Regular Expression Syntax" is used.

Default value: case sensitive

Logging of job-runtime in the database

RUNTIME_INFO_DAYS_OF_DB_STORAGE

This variable specifies the number of days for which runtime information for the jobs processed by d.3 is maintained in the database table `runtime_info`. API-Call, asynchronous jobs and hostimport are saved based on total runtime, hook-runtime, SQL-runtime and time waiting for the file system. If this variable is set to “0”, then no information is logged in the database.

Default value: 0

Enable view displaying watermark

WATERMARK_SUPPORT

By enabling the parameter you can grant users document class rights so that these users are only allowed to view documents with watermarks.

> **Note**
>
> - An installation of d.ecs pdf extension version 1.4.2 or higher is required.
> - The display of watermarks only works if the parameter **V8RunningUpdateBackComp** is disabled.
> - The display of watermarks is only valid for documents which are stored in PDF format.

Default value: 0

Audit-trail

Logging of changes made by users

AUDIT_USER_CHANGES

If this parameter is enabled, then modifications of users in not directly document-related data (e.g. Set Checkout Status or Change Password) in the database are logged.

Default value: No

Audited document types (complete)

audited_doc_types

Short name of the document types for which all modifications in the documents are to be logged in the audit-trail.

Audited document types (basic)

audited_doc_types_basic

Short name of the document types for which all modifications in the documents are to be logged in the audit-trail (whitout details).

Reproducibility of rendering and indexing

AUDIT_PROTOCOL_TIF_OCR

With enabled parameter the rendering and indexing of documents in the database can be exactly reproduced.

Notification in case of unauthorized access

AUDIT_LOG_ACCESS_DENIED

If the switch is enabled, unauthorized document access attempts are logged within the security context.

Key/Value-Cache (jStore)

jStore-bucket

KV_STORE_BUCKET

The name of the bucket (ID prefix) in jStore. If this parameter is unset, the value from parameter 'd3fc_server_id' will be used.

Asynchronous cache warming for imported documents

KV_STORE_IMPORTED_WARMING

d.3 async fills the key/value cache with the most recently imported documents.

> **Note**
> Independent of this parameter the cache is filled automatically by each search query.

Default value: No

Number of most recently imported documents

KV_STORE_ASYNC_IMPORTED_COUNT

This specifies the number of the most recently imported documents which d.3 async writes to the key/value cache. The latest documents are loaded into the cache until the here specified number is reached or no further documents exist in the cache.

Default value: 0

Asynchronous cache warming for imported documents

KV_STORE_HOLDFILE_WARMING

d.3 async also considers mailbox entries when filling the key/value cache.

> **Note**
> Independent of this parameter the cache is filled automatically by each search query.

Default value: 0

Number of mailbox entries

KV_STORE_ASYNC_HOLDFILE_COUNT

This specifies the number of mailbox entries which d.3 async writes to the key/value cache. Mailbox entries are loaded into the cache until the here specified number is reached or no further documents exist in the cache.

Default value: 0

Signature:

Check PDF-files for embedded signatures

SIGN_CHECK_PDF

If this parameter is enabled is, then imported PDF files (hostimport or API import) will be checked for embedded signatures. This requires d.ecs pdf extension (in version 1.2.0.1 or higher).

Default value: No

Default-verifier group for signed documents

SIGN_DEFAULT_GROUP

This parameter specifies the default-verifier group for documents which still have to be signed or whose signatures are not yet fully validated. If the document was imported with an invalid signature status, then the document is automatically imported into the verification of this group.

Default value: “”

Advanced optimizations

Database colums with static querying

STATIC_STATEMENT_COLUMNS

The d.3 system uses prepared statements with bind variables to process a large number of database accesses. This helps the database to work faster as the execution plans do not have to be created all over again, if only the value but not the structure of the query are changed.

However, there are situations, when the best execution plan of the database strongly depends on the passed values. A typical case are columns with a strongly uneven distribution of values, i.e. for example that most values exist only once but some values occur in millions of rows.

In this list, you can enter the columns whose comparison values should be statically written to the SQL queries instead of being used indirectly via Bind-variables.

Format: “table_name.column_name” (SQL wildcards % and _ are allowed in the column name)

> **Note**
> Note 1: The default behavior of d.3 is usually the best option. Please use this option only, if you encounter actual drops in performance that can be traced back to unfavorable execution plans. Remove these entries again if they do not yield any improvement.
>
> Note 2: For some columns, d.3 uses static queries automatically without having to specify these columns here (e.g. if filtered indexes are used).
>
> Note 3: This option affects most SQL queries created by d.3 but does not generally affect the all database access.

Number of the max. filtered document types by d.3 search

MAX_DOCTYPES_DSEARCH_FILTERING

Specifies the number of document types, which d.3 is maximum transmitting during the search to d.3 search.

A restriction of the document type can accelerate the full-text search and raise the precision of the results.

> **Note**
> This effect is especially big, if the number of potential results is reduced by the document type list.
>
> If the number of chosen document types is too big, however, and is no insignificant restriction of the results, the filter in d.3 search can have a negative effect on the search performance.
>
> If the list of document types exceeds the defined value, the filtering applies to document types after the full-text-query on d.3 server.

Activities

Show own activities on the Welcome page

HOME_ENABLE_OWN_ACTIVITIES

0: Own activities will not be displayed.

1: Own activities will be displayed.

Show all activities on the Welcome page

HOME_ENABLE_ALL_ACTIVITIES

0: All activities will not be displayed.

1: All activities will be displayed.

Anonymize user names

TIMESPAN_BEFORE_ANONYMIZE_ACTIVITIES

Default: 30

Always anonymize: 0

Never anonymize: -1

The user names will be anonymized after the period of time specified here when they are displayed.

Document types with minimal log level

DOC_TYPES_LOG_MINIMAL[1]

Short name of the document types with enabled minimal log level.

Events: Import, Import of a new version, property update, link documents and dossiers.

Document type with extended log level

DOC_TYPES_LOG_EXTENDED[1]

Short name of the document types with enabled extended log level.

Events: Status transfer, add and remove favorites, signing

Log level for the linking of documents and dossiers

LOG_LEVEL_LINK

Specifies from which log level the linking of documents is recorded in the history.

Log-Level: 1 = Minimum, 2 = Default, 3 = Extended

Log level for unlinking

LOG_LEVEL_UNLINK

Specifies from which log level the unlinking of documents is recorded in the history.

Log-Level: 1 = Minimum, 2 = Default, 3 = Extended

Log level for starting a workflow

LOG_LEVEL_START_WORKFLOW

Specifies from which log level starting a workflow is recorded in the history.

Log-Level: 1 = Minimum, 2 = Default, 3 = Extended

Log level for finishing a workflow

LOG_LEVEL_FINISH_WORKFLOW

Specifies from which log level finishing a workflow is recorded in the history.

Log-Level: 1 = Minimum, 2 = Default, 3 = Extended

Log level for sending a resubmission

LOG_LEVEL_SEND_HOLDFILE

Specifies from which log level sending a resubmission is recorded in the history.

Log-Level: 1 = Minimum, 2 = Default, 3 = Extended

Log level for forwarding a resubmission

LOG_LEVEL_FORWARD_HOLDFILE

Specifies from which log level forwarding a resubmission is recorded in the history.

Log-Level: 1 = Minimum, 2 = Default, 3 = Extended

Log level for acknowledging a resubmission

LOG_LEVEL_ACK_HOLDFILE

Specifies from which log level acknowledging a resubmission is recorded in the history.

Log-Level: 1 = Minimum, 2 = Default, 3 = Extended

Log level for adding a redlining

LOG_LEVEL_ADD_REDLINE

Specifies from which log level adding a redlining is recorded in the history.

Log-Level: 1 = Minimum, 2 = Default, 3 = Extended

Log level for removing a redlining
LOG_LEVEL_REMOVE_REDLINE Specifies from which log level removing a redlining is recorded in the history. Log-Level: 1 = Minimum, 2 = Default, 3 = Extended
Log level for adding a favorite
LOG_LEVEL_ADD_TO_FAVS Specifies from which log level adding a favorite is recorded in the history. Log-Level: 1 = Minimum, 2 = Default, 3 = Extended
Log level for removing a favorite
LOG_LEVEL_REMOVE_FROM_FAVS Specifies from which log level removing a favorite is recorded in the history. Log-Level: 1 = Minimum, 2 = Default, 3 = Extended

The writing of the history entries follows the following system:

1. There are three different log levels: **Minimal**, **Standard** and **Extended**
   - Each document type belongs to the log level "Standard" by default with the exception of:
     i. **DOC_TYPES_LOG_MINIMAL** (the minimal log level)
     ii. **DOC_TYPES_LOG_EXTENDED** (the extended log level)
     Note: This level can be read in the following form:
     "For a document type, all events are logged which have an event level <= of the log level of the document type."
2. Each action (event) belongs either permanently or configurably to a certain log level
   - For the following actions the log levels are freely configurable (d.3 config parameter):
     i. **LOG_LEVEL_LINK**
     ii. **LOG_LEVEL_UNLINK**
     iii. **LOG_LEVEL_START_WORKFLOW**
     iv. **LOG_LEVEL_FINISH_WORKFLOW**
     v. **LOG_LEVEL_SEND_HOLDFILE**
     vi. **LOG_LEVEL_FORWARD_HOLDFILE**
     vii. **LOG_LEVEL_ACK_HOLDFILE**
     viii. **LOG_LEVEL_ADD_REDLINE**
     ix. **LOG_LEVEL_REMOVE_REDLINE**
     x. **LOG_LEVEL_ADD_TO_FAVS**
     xi. **LOG_LEVEL_REMOVE_FROM_FAVS**
3. The log levels/events are defined as follows:
   a. Minimal log level (so it is always written):
      i. **Import**
      ii. **Update_Client**
      iii. **Update_Job**
      iv. **Update_Server**
      v. **Update_Inherit**
      vi. **ImportNewVersion_Client**
      vii. **ImportNewVersion_Hostimp**
      viii. **ImportNewVersion_Async**
      ix. **StatusTransfer**
      x. **PublishForWeb**
      xi. **ClientEvent** (so specifically events triggered by the d3fc API)
   b. Standard log level:
      Any event that has not explicitly been assigned a minimal or extended log level (but is listed here):

   a. **Import_Hostimp**
   b. **Import_Async**
   c. **LinkToChild_Job**
   d. **LinkToParent_Job**
   e. **AddNote**
   f. **Sign**
   g. **RegisterID**
   h. **UnregisterID**
   i. **DeleteVersion**

c. Extended log level:
   Only if explicitly defined by parameter (see listing 2.1).

4. Exception:
   - Depending on the parameter **`AUDIT_PROTOCOL_TIF_OCR`**, the following events are only logged for "Extended" document classes or not at all (parameter setting to 0 means no logging)
     i. **RenditionJobCreated_001**
     ii. **RenditionJobProcessed_001**
     iii. **RenditionJobCreated_002**
     iv. **RenditionJobProcessed_002**
     v. **RenditionJobCreated_003**
     vi. **RenditionJobProcessed_003**
     vii. **RenditionJobFinished**
     viii. **RenditionJobFlagReset**
     ix. **KeywordsSend_OnChangedType**
     x. **KeywordsSend_OnImport**
     xi. **KeywordsSend_OnClientAPI**
     xii. **KeywordsSend_OnChangedNote**
     xiii. **KeywordsSend_OnTranfer**
     xiv. **KeywordsSend_OnNewVesion**
     xv. **KeywordsSend_OnRenderedOcr**
     xvi. **KeywordsSend_OnChangedAttribute**
     xvii. **KeywordsSend_OnSynchFile**
     xviii **KeywordsSend_OnProcOCR001**
     .
     xix. **KeywordsSend_OnServerAPI**

- If an event of the level MINIMAL occurs (see 3.1.), this is logged in any case
- If an event of level STANDARD occurs (see 3.2.), this is logged for documents which belong to a document class of level STANDARD or EXTENDED (see 1.1.).
  => i.e. conversely, the system does NOT log for documents of a document type that belong to the MINIMUM level.
- If an event of level EXTENDED occurs (see 3.3. / 4.), this is logged for documents which belong to a document class EXTENDED (see 1.1.).

Transport system

Backup of the repository master data
MASTERDATA_BACKUP Enables the automatic backup of the repository master data to be periodically performed by the d.3 async master process. The time and interval are configured via the parameters **MASTERDATA_BACKUP_TIME** and **MASTERDATA_BACKUP_INTERVAL**. The backup file is stored in the document tree under the directory **config\d3admin\mdbackup**.
Master data backup interval

MASTERDATA_BACKUP_INTERVAL

The master data backup will be done within fixed intervals. Valid values for **MASTERDATA_BACKUP_INTERVAL** are:

T: daily

W: weekly (on every Monday)

M: monthly (use **MASTERDATA_BACKUP_DAYOFMONTH** for setting up the day of month)

With **MASTERDATA_BACKUP_TIME** the time is specified, when the backup is to be executed.

Time stamp for the backup of all repository master data

MASTERDATA_BACKUP_TIME

Specify a time in the format "hh:mm".

Day of the month

MASTERDATA_BACKUP_DAYOFMONTH

Specifies the day of the month on which the backup is created. This only has an effect, if the parameter **MASTERDATA_BACKUP_INTERVAL** is set to "M".

E-mail notification

MASTERDATA_BACKUP_SENDMAIL

Enables the notification to be sent after the backup has been created.

"0": Do not notify

"1": Notify in case of errors

"2": Always notify

The parameters **SMTP_SUPPORT** and **SMTP_GATEWAY** must be set.

E-mail address

MASTERDATA_BACKUP_MAILADDRESS

E-mail adress for notification after backup.

Automatic import of repository master data files

MASTERDATA_IMPORT

Enabling this parameter you can configure the d.3 async master process to administrate repository master data such as users, user groups (e.g. create or update users) via control files/interface files. The automatic import is executed every 5 minutes.

Adoption of the storage settings for document type changes

MASTERDATA_TRANS_DTYPE_STORAGE_SETTINGS

By enabling the parameter the storage settings are overwritten for document type changes which are imported via transport files.

This affects the following settings:

- Time to live -> In months

- Time to live -> From

- Time to live -> Outsource to secondary storage

The settings "Time to live -> Event-based deletion" is not considered and thus is always transported.

More rules apply:

- When creating an new document type or changing a document type this parameter has no effect.

- Each change to the settings for the document types is recorded for the transport system and is written to the transport files.

- If a document type is transmitted to another d.3 repository via a transport file but the document type already exists, then the storage settings for disabled parameters will not be overwritten.

Adoption of the caching settings for document type changes

MASTERDATA_TRANS_DTYPE_CACHE_SETTINGS

By enabling the parameter the caching settings are overwritten for document type changes which are imported via transport files.

This affects the following settings:

- Retention time in cache -> In days

- Retention time in cache -> Allow removal ahead of schedule on lack of disk space

More rules apply:

- When creating an new document type or changing a document type this parameter has no effect.

- Each change to the settings for the document types is recorded for the transport system and is written to the transport files.

- If a document type is transmitted to another d.3 repository via a transport file but the document type already exists, then the caching settings for disabled parameters will not be overwritten.

d.3 presentation server

d.3 presentation server- hostname

PRESENTATION_SERVER_HOST

Hostname of the machine running d.3 presentation server.

d.3 presentation server - TCP/IP port

PRESENTATION_SERVER_PORT

TCP/IP Port under which d.3 presentation server is accessible.

Default value: 8081

d.3 presentation server - HTML path

PRESENTATION_SERVER_PORT

Contains the context path for d.3 presentation server.

d.3 presentation server- encryption

PRESENTATION_SERVER_SSL

If d.3 presentation server works with SSL, this parameter must be enabled.

d.3 presentation server- security level

PRESENTATION_SERVER_SECURITY_LEVEL

The security level determines which security features are used by d.3 presentation server and the applications.

0: The new security features will not be used.

1: The new security features will be used.

d.3 presentation server - expires time for URLs

PRESENTATION_SERVER_EXPIRES_SECS

If the security level is at least 1 the expires time of generated URLs by d.3 server is set to this value. The time is specified in seconds.

Default value: 60

d.3 presentation server - mapping of d3fc hosts to presentation server URIs

PRESENTATION_SERVER_MAPPING_D3FCHOST

Here the D3FC hosts are to be entered which are being assigned 1 to 1 (index number) to the corresponding PS-URIs.

(**PRESENTATION_SERVER_MAPPING_PSURI**)

d.3 presentation server - assignment presentation server URIs to d3fc hosts

PRESENTATION_SERVER_MAPPING_PSURI

Here the PS-URIs to the respective D3FC hosts are to be entered.

Example: https://ps-server:8444

(**PRESENTATION_SERVER_MAPPING_D3FCHOST**)

> **Note**
> For this mapping to take effect, no host must be specified in the d.3 config for **PRESENTATION_SERVER_HOST**.
>
> This means, if no **PRESENTATION_SERVER_HOST** is defined, and "**d3_host**" is passed as an import parameter of the call, then this presentation server is returned. However, if none is passed, another mechanism applies and the server returns itself as the PS-host.

d.3 presentation server - Enables the HTTP interface of d.3 server

ENABLE_HTTP

Determines whether the HTTP interface of the d.3 server is enabled.

Configuration using the d3addon.ini

Some special parameters cannot be set using d.3 admin. These parameters can only be manually defined in the file **d3addon.ini**.

All parameters can be found together with a short description in the **d3fixed.ini**.

Having modified the **d3addon.ini**, you must restart the d.3 processes (d.3 server, d.3 hostimp and d.3 async) for the new configuration to be read properly.

> **Warning**
> Do not change the entries in the **d3fixed.ini**! Changes here can result a standstill of the d.3 repository and to loss of data.

> **Note**
> The capitalization is significant.

If you create a new repository with this version, then this repository is further defined with the following entries in the **d3addon.ini** file:

These parameters must not simply be set manually in an existing system, as they require according data structures which might have to be created during a migration.

This is also the reason, why those parameters are not all included in the **d3config.ini**.

Parameters configured in the **d3config.ini** as well as in the **d3addon.ini** require no migration but must not be set during updates as this might lead to faulty behavior, e.g. **REVOKE_PROCESSING_ADMIN**. For new installations these parameters can be set without risk as the new behavior is then usually desired.

In the following, all parameters which are preset in the **d3addon.ini** for new installations are explained.

CONFIG_PATH_D3CLIENT = "C\:\\d3\\D3T.dok\\config\\d3client"

Path to the directory containing the search templates as well as the client profile files (client config).

UTF8_SUPPORT = "1"

Enabling the unicode support of the d.3 repository.

SEARCH_DEP_DOCS_IN_DB = "1"

Information on the dependent files is read from the database and the information is managed here.

SUFFIX_DEP_DOCS_ARCHIVE_THREE_DIGITS = "1" see the information on the old identifier **SUFFIX_ABH_ARCHIVE_DREIZIFFRIG** in the **d3fixed.ini**.
TURBO_DOC_FLAGS_2 = "1" Document flag information for the acceleration of the d.3 document search
SUBDIR_VERIFICATION = "1" Create/use sub-directories in the d.3 doc tree for the verification directory
SUBDIR_ARCHIVE = "1" Create/use sub-directories in the d.3 doc tree for the archive directory
SUBDIR_NOTES = "1" Create/use sub-directories in the d.3 doc tree for the notes directory
SUBDIR_WFL_LOG = "1" Create/use sub-directories in the workflow-protocol directory
REVOKE_PROCESSING_ADMIN = "1" Allowing to revoke the processing version by an administrator
MODIFY_ATTR_DB_POSITION = "0" Allowing/prohibiting the retroactive change of the database position of a property in the document type
ENABLE_DUMMY_WRITE_ACCESS = "0" Withdrawing the write-permissions to dummy-document type
VALIDATE_NUMERIC_FIELDS = "1" Enabling the verification if a numeric property only contains numeric characters.
OVERWRITE_NEW_ARCHIVE_PARAMS = "0" Internal parameter specifying if the parameters for the initial creation were already written to the d3addon.ini.
WORKFLOW_NOTIFICATION_AUTO_ACKNOWLEDGE_ON_LOCK 1: 1: With the parameter you can control whether notifications from other users for a workflow step are to be automatically acknowledged. These notifications are no longer visible in the users' mailbox, if a user opened a workflow step for editing which is then locked. 0: If the notification has been sent via a group, then it is assigned to the caller and is then no longer visible for the other users.
D3_SERVER_HOST Via the d.3 server config variable **D3_SERVER_HOST** the host name under which the d.3 server is externally accessible in cluster configurations can be specified. By this the default host name (**d3fc_server_name**) can be overwritten. This host name is then used for the creation of all workflow URLs. This affects the API functions **GetPSUrl**, **LockWorkPathStep**. This config variable can be set in the **d3addon.ini**.

1.4.5. System monitoring

Below the system monitoring, you will find d.3 async job inspector and the runtime information.

d.3 async job inspector

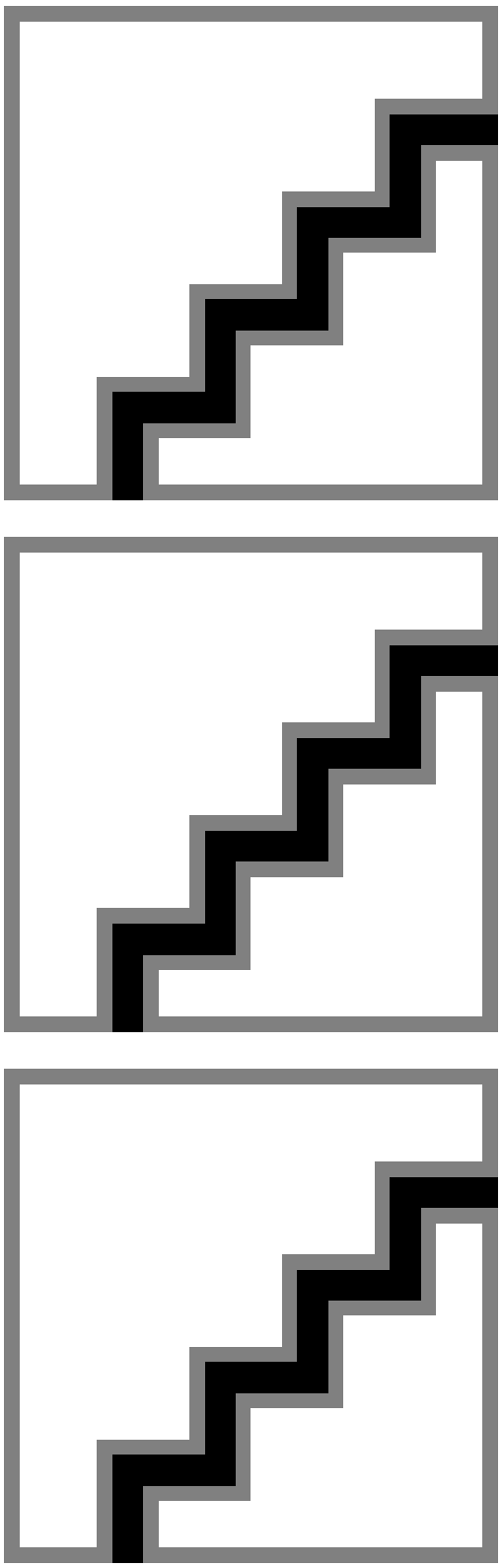
d.3 async job inspector allows to monitor and control the d.3 async jobs being executed on the d.3 server in the background.

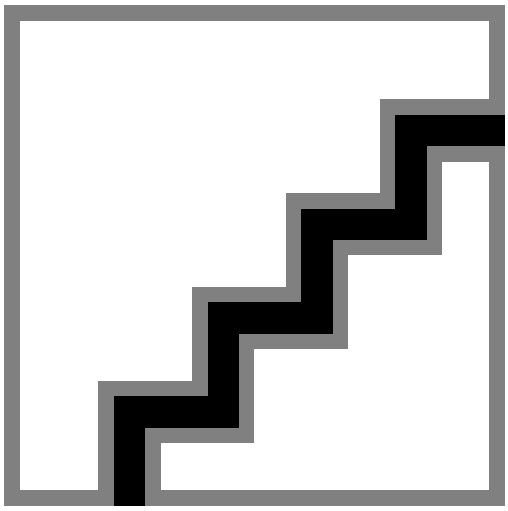
This extension of d.3 admin is a valuable help for the analysis of problems (e.g. in dossier link). You can use d.3 async job inspector to repeat and delete jobs or to change their priority.

For additional information on managing d.3 async jobs read the manual d.3 admin async job inspector.

Runtime information

Via **System monitoring > Runtime information** in d.3 admin you can get an overview about the various runtime information of the last days/hours of a repository.





Elements:

The runtime information consist of the following display elements:

- Job overview:
  A listing of the various job types is on the left side. Here you can read for every job type the total number for the transmitted results (calls) within the selected time frame (timeline).
- Timeline:
  A horizontal dropdown list on the upper margin. The displayed time is equivalent to the maximum period of time of the recorded data (see parameter `RUNTIME_INFO_DAYS_OF_DB_STORAGE`). Furthermore the currently selected period of time can also be read.
- Time chart:
  The selected data are visualized in charts below the timeline. Within the charts are various curves (Total time, SQL time, Hook time, Filesystem time, Service time and K/V.Store time) available for selection, whereby initially the total number is displayed respectively and the other curves can be additionally displayed. The first (upper) chart is the main chart where you can find the button **Add Chart** to add further charts.

Operation/procedure:

In the initial view of the runtime information the information of all jobs of the full period of time is visualized in the main chart.

To filter for certain criteria on the information to be displayed, several engaging mechanisms are provided:

- In the job overview a specific job type can be added to the view filter while a renewed click deletes the job type from the filter. The view filter may contain any number of job types.
  If the view filter contains at least one element (job), then this is active and all not contained job types are greyed out or are no longer displayed in the time charts.
- The timeline allows you to select a time span using the mouse (press and hold left mouse button, drag and drop).
  The display of events in the job overview adapts to the selected time span. Additionally, the time charts zoom into this span while the resolution of the displayed information is also increased in dependence of the selected time span to provide more information in this detailed view. The time and runtime axis of the time charts scale respectively to the information to be displayed automatically.
- Within the main chart the selected data range can be further refined (or the refinement can be undone) by using the mouse wheel. The same behavior as for the (rougher) selection via the timeline is given here.
- Within the timeline as well as in the main chart the selected time span can be moved by the mouse (press and hold the left mouse button).

Via the button **Add Chart** in the main chart you can create up to five additional time charts. These appear directly below the main chart while the previously created charts are moved downwards. The visualized information are equivalent to the criteria defined by time and job selection. Also the zoom of the main chart has effect directly to these additional charts.

The charts created by this way have two buttons:

- A button with a minus character: With a click on the respective time chart it can be hidden. In the hidden status there is only the minus button (now with a plus icon instead minus) and the **Remove** button.
  A renewed click on the button shows the chart again. Previously made settings within the chart remain.
- A **Remove** button: By clicking on the respective time chart it can be Removed. By doing so, the respective time charts disappears together with its buttons and the respectively made settings. The charts below move upwards.

Within each individual time chart the desired timeline is optionally combinable and can be shown and hidden. The timeline "Total time" displays just the sum of the specific times.

Every displayed data point within the charts displays one or multiple summarized events. For the timeline "Total time" a tooltip with detailed information regarding the causing job types (is only one job type involved, then this is directly displayed) and the underlying calls appears. For the remaining timelines the exact time value of the respective type of time appears as information.

1.4.6. More services

More services are also d.3 abo service, d.3 search, d.ecs rendition service and the Form assignment.

d.3 search

d.3 search is used for the full-text search in the d.3 repository. Detailled information on this can be found in the corresponding documentation.

d.ecs rendition service

d.ecs rendition service is used to convert documents on the server and is described in its own documentation .

Form assignment

The form assignment refers to the process portal and is described in the own documentation process portal and form assignment.

1.4.7. Webhooks

In this chapter, you can find information about webhooks in d.3 admin.

Working with events

Webhooks let you respond to events that are triggered by specific processes in a repository. You can also use the HTTP responses to your webhooks to control the performance of the processes.

To configure a webhook, you must provide a URI. This URI is called when the event occurs. You can configure multiple webhooks for one event. These webhooks are called one after the other when the event occurs.

Enter the configurations of the webhooks in the **Webhooks** menu item for your desired events.

Webhooks can be called for a series of events that may occur while working with documents in a repository.

- Search

- Create documents or dossiers
- Create a new version
- Update properties
- Delete documents or dossiers
- Link items
- Change status
- Create a generated document

When the event occurs, an HTTP POST request is sent to each entered URL. Based on the selected event, various information (about a stored document, for example) is sent to the webhook in the body of the HTTP POST request. The webhook can use this information as the basis for starting further actions.

The information objects for each event are described below.

Search

These events are triggered when users perform a search in the repository.

- Event 1: Before searching

The event is called before a search. The transferred search criteria have not yet been checked for plausibility at this point in time.

- Event 2: After searching

The event is called at the end of the search directly before the results are delivered to the user.

Create documents or dossiers

These events are triggered when a new document or dossier is stored in the repository.

- Event 1: Validate storage dialog

The event is called when properties have been assigned to a newly imported document.

- Event 2: Before creating a document or a dossier

The event is called before users create a document or dossier. You can change the values of the advanced properties in the webhook.

- Event 3: After creating a document or a dossier

The event is called after users have created a document or dossier. The process is already complete during this event.

Create a new version

These events are triggered when a new version of a document is stored in the repository.

- Event 1: Before creating a new version

The event is called before users create a new version of a document. During this event, the system checks whether the document already exists in the repository. You can still change the advanced properties at this time.

- Event 2: After creating a new version

The event is called after users have created a new version of a document. The process is already complete during this event.

Update properties

These events are triggered when the properties of a document or dossier are changed.

- Event 1: Validate properties

The event is called when the document properties are validated.

- Event 2: Before updating properties

The event is called before the properties of a document are updated. You can change the values of the advanced properties in the webhook.

- Event 3: After updating properties

The event is called after the properties of a document have been updated. The process is already complete during this event.

Delete documents or dossiers

These events are triggered when a document or dossier is removed from the repository.

- Event 1: Before deleting a document or a dossier

The event is called before a user deletes a document or dossier. At this time, the system has already checked whether the user is permitted to delete the document.

- Event 2: After deleting a document or a dossier

The event is called after a user has deleted a document or dossier. The process is already complete during this event.

Link items

Events with the **Link items** type are triggered when two items are linked to each other. The event is triggered with the following links when:

1. A document is linked to a dossier.
2. A document is linked to another document.
3. A dossier is linked to another dossier.

- Event 1: Before linking items

The event is called before the two items are linked.

- Event 2: After linking items

The event is called after two items have been linked. The process is already complete during this event.

Change status

These events are triggered when the status of a document is changed.

- Event 1: Before changing the status

The event is called before the document status is changed.

- Event 2: After changing the status

The event is called after the document status has been changed. The process is already complete during this event.

Create a generated document

These events are triggered when a new dependent file is saved for a document. In a repository, this event is triggered by the automatic generation of a stable long-term PDF image of the document.

- Event 1: Before creating a generated document

The event is called before a new dependent file is saved.

- Event 2: After creating a generated document

The event is called after a new dependent file has been saved. The process is already complete during this event.

Configuring webhooks

You can edit, add or delete the webhook entries assigned to the respective events.

Webhooks are called one after the other when the event occurs. You can change the processing order by choosing **Edit order**. Choose **Position** to move a selected webhook entry within the list.

Choose **Refresh** to reload the webhook configuration from the server.

You can use **Export** and **Import** to export and import the webhook configurations. The configuration data is transferred as a **JSON** file.

Enter a name for the webhook as the configuration ID. The ID of the webhook is generated automatically and cannot be changed.

The **URI** defines the web address that is called when the event occurs. The system supports absolute HTTP addresses and relative paths for local web addresses. When you make the entry, the system performs a simple plausibility check of the URI but does not check the connection.

Choose **Enable webhook** to enable the webhook in the DMS system.

Evaluating the response to the webhook

You can define how and whether the HTTP response to your webhook is to be evaluated by the DMS system.

Use **Do not wait for response** for webhooks that do not have to affect the running process. The webhook is called asynchronously. The DMS system does not wait for an HTTP response. The duration of the process is unaffected.

Use **Consider status code** for webhooks that may terminate the process. The webhook is called synchronously and the system waits for the HTTP response. The DMS system evaluates only the status code of the HTTP response. The running process cannot be terminated with the status code. For this purpose, the webhook must return a status code in a range between 400 and 499.

Use **Consider status code and evaluate the content of the response** for webhooks that terminate or change the process. The webhook is called synchronously and the system waits for the HTTP response. The DMS system evaluates the status code and content of the HTTP response.

Restriction to a specific category

Use **Execute only for this category** to restrict the execution of the webhooks to one category. If you do not select a category, then the webhook is called for each category.

Defining an authentication key

The **API key** option lets you define an authentication key. This authentication key is also sent during the HTTP call of the webhook. As a result, the webhook can check whether the HTTP call actually originates from the DMS system. Authentication keys are sent only if you have specified a relative URI as the URI.

Select the asterisk (*) next to the input field to display the API key as plain text.

> **Note**
> With the latest version, "Authorization: Bearer APIKEY" is now transferred to the webhook as a header instead of "x-dv-api-key: APIKEY".

Controlling performance

Use **Timeout in milliseconds** to define how long the DMS system waits for the HTTP response to the webhook. The HTTP request is terminated once the timeout ends.

Enable the timeout to be repeated with **Recall webhook up to three times in case of a timeout or an incorrect response**. If a timeout occurs, the HTTP request is performed again. The HTTP request is sent a maximum of four times before it is permanently discarded.

Information fields

The **ID** text field displays the internal ID of the webhook. The **Entry point** text field displays the internal assignment of the webhook to an event.

1.4.8. Dataset webhooks

In this section, you can learn how to configure dataset webhooks.

What are dataset webhooks?

A dataset is a list that you can use to select property values. When you assign a dataset to a document property, users can then select only values from this dataset.

The dataset webhook is a configuration that you can use to dynamically load a dataset range from a web address.

Working with dataset webhooks

You can edit, add or delete entries from the list of dataset webhooks.

You can use **Export** and **Import** to export and import the configurations. The configuration data is transferred as a **JSON** file.

To use the dataset webhooks, go to **Advanced properties > Assignment of datasets** and select **Webhook** as the dataset. Select a name as well.

Data types

When you create a webhook dataset, the data type **Dynamic query (webhook)** is selected automatically. You cannot change the data type.

Defining the webhook values

Enter a name for the webhook as the configuration ID. The webhook ID is generated automatically and cannot be changed.

The **URI** defines the web address from which the dataset is loaded as a resource. The system supports absolute HTTP addresses and relative paths for local web resources. When you make the entry, the system performs a simple plausibility check of the URI but does not check the connection.

The **API key** option lets you define an authentication key. Select the asterisk (*) next to the input field to display the API key as plain text.

Use **Timeout in milliseconds** to define how long the DMS system waits for the HTTP response to the webhook. Once the timeout ends, the HTTP request is repeated and then terminated after a maximum of three failed attempts.

The **ID** text field displays the internal ID of the dataset webhook as information.

1.5. General concepts

In the following you will find information on the topics, among others:

- Structure of the electronic d.3 repository
- Case study
- Advanced properties fields
- Validation
- Assignment of datasets
- Using hook functions
- Concept of access rights
- Notes on the transport system

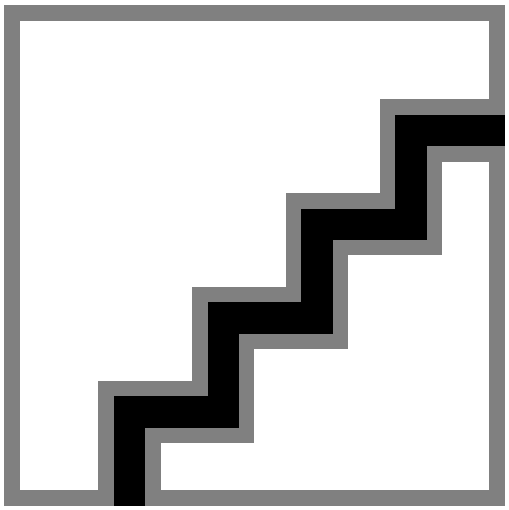
1.5.1. Structure of the electronic d.3 repository

Basically, the d.3-Repository (formerly known as "Archive") is designed to resemble a typical filing cabinet. The filing cabinet is also organized in a specific way in order to quickly find dossiers and documents. You can classify documents in drawers and on shelves by document types or dossiers and, thus, find them quickly.

In a filing cabinet, however, the documents can only be retrieved according to one characteristic: the physical location. This is different from the d.3 repository. Documents are classified with several freely definable properties used to search the related documents. These properties can be, for example, the date or the name of the document.

Some of these properties define the storage location in the repository (a kind of electronic archive), which is determined by a category (document type or dossier type).

Securing against unauthorized access in the d.3 repository is realized in a similar way as with a filing cabinet.



A fixed location does not exist in a d.3 repository. On the contrary, it is possible to store one document in several logical dossiers, without keeping several redundant copies of it in d.3. Thus, you could store and

find your invoice in an order folder as well as in a customer folder. This way you avoid redundancy: There are references to the same document in various dossiers.

The manual distribution and mailing within your organization become redundant. Once electronically archived, all documents are immediately available to all eligible users and can also be used in a business process, if applicable.

1.5.2. Case study

The following case study will exemplify some of the opportunities which d.3 offers for the design of a repository:

In a company, orders, the respective order confirmations and invoices are to be sored in an order folder. Several order dossiers of one supplier should be kept together in a supplier dossier.

Thus, the folder structure looks like this:

An order folder is to be created when an order is imported, which should be stored with the supplier name and the order date. The order dossier is linked with the supplier dossier. If this does not exist, yet, it is created.

The supplier folder contains the supplier name, the address and customer number. This information is retrieved from a database table according to the supplier name.

The order confirmation includes a supplier name and the date. The invoice additionally contains the invoice amount. Thus, the following properties are necessary:

- Supplier: Alphanumeric value, 35 digits
- Address: Alphanumeric value, 35 digits
- ZIP City: Alphanumeric value, 35 digits
- Customer number: Alphanumeric value, 35 digits
- Date: Date value
- Invoice amount: Numeric, 0.00 - 9999999.99

1.5.3. Advanced properties fields

The freely defineable properties are initially created for all document- and dossier types and collected.

By using the same advanced properties fields in different document- and dossier types, related documents can also be found disregarding their document type.

Thus, you could find the customer order, the according invoice and other related documents based on the “order number".

1.5.4. Validation

For some advanced properties fields it is necessary or desirable that the entries follow certain rules.

Let us suppose, the entry in the property "order number" shall follow certain rules so that only entries in upper-case letters followed by seven digits are permitted.

In this case a so called "regular expression" can be specified, which only allows the entry of this desired pattern.

A "regular expression" is something like a template which only permits entries of the specified form and rejects others as incorrect.

The definition of "regular expressions" uses a language which will explained below in this manual.

For the above mentioned example you could use the following “regular expression” to define the desired rule for the entries:

```
[A-Z][0-9]\{7\}
```

1.5.5. Assignment of datasets

In some cases, only a limited selection of fixed values should be used for an advanced properties fields. The users can/must select an property value from a picklist (when searching or importing).

Suppose you capture correspondence in a document type and decide, if it is incoming or outgoing mail based on an property "In/ Out". The allowed values for the property "In/ Out" are therefore "incoming" and "outgoing".

The solution to the problem is to use a dataset from which a value must be selected. This dataset can either be created by the d.3 administrator, provided from a database table of the d.3 repository or be retrieved by a newly provided function. By calling a hook function, the dataset can also be read from an external table.

In the first case a dataset can be created in d.3 admin. In the second case, an additional database table can be created in the d.3 database, from which d.3 obtains the pool of values.

The values can also be retrieved from a database view. In this case, the database view may only return one column.

> **Note**
> The dataset can only be defined as a suggestion or default for a suitable advanced properties fields. If then the field in the application has to contain values from the list or if additional entries are allowed, will be defined in the properties of the properties of the document type.

> **Note**
> You can use a dataset webhook to dynamically load a dataset range from a web address and assign it to a property field. A webhook from the list of configured dataset webhooks is assigned to the **Webhook** dataset range.

1.5.6. Using hook functions

Let us suppose, the “order number” of the document type “supplier order” is to be assigned automatically by d.3, whereby the first two characters should depend on the person who accepted the order and the following digits should be a counter.

To automatize this transaction, a hook function can be used. A hook function is a function executed on the d.3 server triggering a customer-specific action.

> **Note**
> Hook functions are developed on request (customer-specific). Before developing your own hook functions you are urgently requested to attend a training with the d.velop AG.

1.5.7. Concept of access rights

The concept of access rights in d.3 defines "Who may access What in Which way".

In the concept of access rights you determine about the document classes the What, e.g. a subset of all documents of a document type. Example: Only documents with an certain accounting range.

Concept of access rights common

The kind of access, like the Which way, can include the following manifestations (partially still extendable):

Read

Write

Status change

Links (manual) of documents in a dossier or another document.

You can select several options to all rights:

Right assigned: Prerequisites for an user to access to a document

Right denied: Explicit denial of a right, cannot be overridden at all

Right ignored: Right will neither be assigned nor be denied; will no decision be adjusted on other levels, the right applies as denied (implicit denial of the right)

The concept of access rights allows to define restrictions via properties which are themselves combined in authorization profiles (former roles) to permit access based on a certain job profile. However, permissions can also be explicitly withdrawn.

Furthermore, the access rights concept designates which users may work with the repository at all (policies).

Authorization profiles and document classes

In the concept of access rights from version 7, the **authorization profiles** and **document classes** are used to grant permissions. Like the user groups, the authorization profiles are used to combine several document classes and to define the access rights to the specified class.

Authorization profiles received a accumulation of document classes, which employees need for their tasks.

A document class reference a certain document type or all document types combined in the authorization profile globally. If a document class refers to a certain document type, you can configure the access limitations using the properties of this document type.

You can imagine a class as a template placed on a (or across all) document type(s) filtering using certain criteria. With document classes you determine subsets of all documents, several templates can be combined in a profile.

Example: The document type “Invoices” is defined in your d.3 repository. This document type has ten properties. One of these properties is the field named "Invoice amount" of the type "Numeric". If you

now create a document class with the document type "Invoices", then you could enter the value "-5000" in the field "Invoice amount". This value means: All invoices up to an invoice amount of 5,000 (currency units!).

Descriptive text for the document classes:

- A document class is a dynamic set of d.3 documents.
- A document class is defined by search criteria in d.3.
- By default, every document type is a document class, as a default-document class is automatically generated together with each document type. The class has the same name as the document type. The class ID of the default-document classes begins with the phrase CLAS_. Additionally, the names have the suffix (C). For German d.3 installations the suffix (K) is appended.
- A document class usually is a sub-set of a document type. However, you can also create document classes across all document types.

Examples for document classes

- All invoices (default class)
- All invoices with an invoice amount of more than 100000.
- All orders by supplier "Müller"
- All item numbers like S30854-%-123_-%
- All drawings of project number XY4711
- all documents of a company location
- all documents of a client

Authorization profiles

- An authorization profile is a summary of combinations of document classes and associated rights.
- An authorization profile can be the technical counterpart to the job profile of a group of employees (a department etc.), i.e. all information is compiled to which, say, a sales clerk for Europe must have access.
- Since employees can be assigned to multiple profiles, the permissions could be extended e.g. for managerial users.
- Furthermore, you can assign "global" profiles and withdraw some rights using additional profiles (negative rights).
- An authorization profile thereby defines an access right for a certain range of documents. This allows that users can be assigned to authorization profiles, by which they receive rights to the documents.
- A user can be assigned to an unlimited number of authorization profiles.

> **Note**
> For a more flexible handling, direct combinations of document classes and rights can be assigned to users. However, this should only be done to handle exceptions, e.g. to quickly block access to documents.

Example: Authorization profile of project staff for project XY

Document class - right

Invoices for project XY with amount lower than 10,000 - Read released documents

Drawings for project XY - Status change verified

Correspondence for project XY - Edit (content and properties)

Rights

Permissions are configured for every document class within a authorization profile. The following rights can be assigned to a document class:

- Read
- Write (Create, Delete, Edit, Versioning)
- Status change (Verification, Verified, Release)
- Creating and remove links

Every permission can also be used as a negative right (denial of this right). An explicit negative right can be assigned on the level of user <->document class or when assigning a class to a authorization profile. It acts as a type of "emergency brake". If a user is explicitly denied the permission for a class, then the right iss denied even if he was granted the permission in his authorization profile.

> **Note**
> An explicit denial of a right can NOT be overridden by granted rights over other assignments!

There are two ways to grant rights to a document class to a user:

- Implicit: You assign one or more authorization profile(s) to a user.
- Explicit: You directly assign individual document classes to a user.

Document types and the class concept

Document types available for a user in d.3 smart explorer and in d.3 import:

In order to find out which document types a user should be offered one must first collect all document classes he should have rights for. These classes are related to one or more document type. Thus, a list of document types is the result, which is available to the user.

Furthermore, there is a difference, if the user has reading or writing rights to the document type. To get a document type in d.3 import (manual storage), the user needs writing rights to the document class, from which the document type was derived.

If a user was only assigned to document-type-independent classes, he will not be able to see any document types in the d.3 applications (d.3 smart explorer, d.3 import) at first. In order to provide the desired document types to this user, he is assigned with the default classes without specified rights via the authorization profile or directly (see also chapter Document classes).

Effective rights of a user

In order to find out about the effective rights of a user for the document classes all authorization profiles of the user have to be checked first. Then, the user's rights directly assigned to classes are laid on top of this.

> **Note**
> If a user gets via the assigned authorization profile and the immediately assigned document classes different rights, then "wins" that assignment, that was immediately assigned. Even a denied right can be overridden by this!

A document can be assigned to no, one or many classes. A user will be granted access to a document, if the document is assigned to at least one class, for which the user has the desired right.

Basically you must make sure to assign the right at least at one point or he will not be able to find any document. The default setting for "Right ignored" is always "Right rejected"!

1.5.8. Notes on the transport system

Projects and milestones - the whole purpose

To do changes on the master data in d.3 admin, you have to enable the edit mode for a project. Thus, all changes in the active milestone of the project are recorded.

During an export of the project or the milestones of the projects all changes are included that were done within the project. Exactly these changes can be transmitted to another d.3 repository. If there are collisions by changes to the same master data within several projects, the state which is current when closing a milestone, is determined and saved. This considers all dependencies (e.g. for changes to document types the advanced properties of the document types are also be saved).

Projects common

While working with d.3 admin, keep in mind that changes only can be done in the edit mode. To get into the edit mode, you need at least one project, which is in the status “open”. For each project only one administrator can switch into the edit mode. Several administrators can work in different projects at the same time, but never in the same project.

Project creation

For the creation of a project a name (max. 50 characters) must be specified. This name must be unique and cannot be changed just like that. Optionally a description (max. 255 characters) can be entered. The project is created in the status "opened" and cannot be closed or deleted by the administrator.

Renaming projects

Projects cannot be renamed via the user interface of d.3 admin. To do so, you have to use the plugin "d.3 project manager".

Closed projects

There are also closed projects, that are not displayed by default. Though it is either a created snapshot project automatically created by the system or projects created while importing transport files. These projects cannot be opened anymore.

Snapshot project

The snapshot project contains backups of the master data which are created automatically when starting an import-process of a transport file. A new milestone is created for each backup.

Open sessions of projects

As mentioned before, for each project only one administrator can switch into the edit mode. Who is working on which project can be seen under "Open sessions" in d.3 admin. If a project is blocked for a longer time, the open session of this project can be closed. To do so, execute a right-click on the session entry. A context menu with the menu option "Close session" appears. Click on the button to close the session. However, a session can only be closed, if at least for an hour no activity has taken place in this session.

Milestones common

Milestones are used to record certain changes in d.3 admin and to export and e.g. to import these to other systems.

A milestone is always a part of a project and cannot be created independently by yourself. Also, a milestone cannot be deleted.

While creating a project, a milestone will be created automatically. Every milestone has a number, which begins at 1, for each project. Milestones are either in the status “opened” or “closed”. Only if a milestone

is in the status “opened” (it is marked as red in the milestone overview), changes are recorded in it. A milestone in the status “closed” (it is marked as green in the milestone overview), it cannot be opened again. For each project a maximum of one milestone in the status “opened” is possible. Though, this is always the last one created. There are also projects, in which none milestone is in the status “opened”. Changes cannot be recorded anymore in such projects. Such projects are created by the import of transport files from another d.3 repository.

Recording data

The following data is recorded within a milestone:

- Datasets
- Document types
- Advanced properties
- Multiple languages
- Dossier generation
- Groups (except memberships of users)
- Document classes
- Authorization profiles
- Authorization profile assignments (not for users)
- d.3 organizational structure
- d.3 policy manager (not for users)
- d.3 restriction set manager (not for users)
- d.3 config (not for all parameters)
- d.ecs rendition service assistant

Changes on master data objects in a milestone

Changes on master data objects are noted within the milestone. Technically viewed this happens:

- Based on the synchronization ID the time of the change is updated
- For every change an entry is created in the database. This entry will also be displayed in the protocol of d.3 admin.

Closing a milestone

If you close a milestone (only possible in edit mode), the following actions will be executed:

- You will be prompted to enter a descriptive text for this milestone.
- By means of the list of changes on master data objects all dependencies will be determined and the current state of these objects will be written into a transport file.
- Attention: Because the current status of an object (e.g. a document type) is used, changes are considered which were made in other projects!
- Only the master data objects are written into the transport file, which have been changed since the last milestone.
- This transport file will be imported into the d.3 repository (using master async into the document type "`Audit-Log`").
- The current milestone will be closed and a new one is created in the status “opened”.

Deleting master data objects

When deleting master data objects, e.g. a document type there is a specialty: The actual files of the object are deleted as before. But the data which are relevant to the transport system, e.g. the synchronization ID are not deleted. This means: If you create a document type with the same short name after deletion again, then no new synchronization ID will be created but the previously used document type short name will be used again.

This may have an effect on the transport of data to another system: If you create a document type in the destination system which has the same short name of a document type of the source system and you delete this in the destination system, a transport from source to destination system will fail because the synchronization ID are not the same.

If you delete master data objects with user reference on the source system, before importing the transport file which contains this change, all references to this master data object need to be removed because such references are not considered by the transport system.

Example: On the source system an authorization profile is to be deleted. This authorization profile exists on the destination system and is assigned to individual users. To be able to execute the delete process via the import of the transport file on the destination system, the user assignment need to be removed on the destination system. Consider also the settings of the LDAP-configuration.

Terminology

With d.3 version 8.0 there not only optical changes in d.3 admin (see chapter Starting d.3 admin) but also some new menu options and terms, which meaning is described here.

Project

A project is a collection of milestones and is used to track changed to repository master data such as document types or advanced property fields as well as changes to configuration parameters. A project has a name and an optional descriptive text. Every project has a status: Either “open”, which means that changes within this project can be recorded or “closed”, which prevents the recording of changes. Once a project is closed, it cannot be opened anymore and is by default not displayed in d.3 admin.

Milestone

In a milestone changes to repository master data respectively a configuration parameter are recorded and can be transmitted to another d.3 repository. Milestones are numbered consecutively beginning with 1 and have the name "Milestone <number>". Every milestone has a status. Either “open”, which means that changes within this milestone can be recorded or “closed”, which prevents the recording of changes. Once a milestone is closed, it cannot be opened anymore.

Import

With the import function recorded changes of projects or milestones as well as backups respectively snapshots can be imported into the d.3 repository. Before each import a backup is done whereby a milestone will be created in the snapshot project.

Export

Via the export function selected projects or milestones can be exported in form of ZIP-files from the d.3 repository. These files can be transmitted via the import function to another d.3 repository.

Backup

A backup contains the complete masta data and configuration parameter of the d.3 repository and can be used as a rollback of an existing or a rebuild of a new d.3 repository. A backup can be created automatically or explicit. The created file has the file extension ".d3backup".

Snapshot

A snapshot contains the same data as a backup and is created automatically by an import process. Snapshots can be managed in a own project with the name "Snapshot", where the project has the status “closed”. Every snapshot is a milestone within the snapshot project.

Transport

A transport is a file, that is created by the export function and its file extension is `.d3trans`. A transport can be transmitted via the import function to another d.3 repository. The transport is intended for the continuous exchange between test- and productive system while the backup is for actual data backup.

Edit mode

In this mode changes to master data und configuration parameter in d.3 admin can be executed and recorded. For each project only one administrator can switch into the edit mode.

View mode

In this mode no changes can be done. All modules within d.3 admin will be started in a readonly mode.

Open sessions

If an administrator switches into a project in edit mode, a session of this project is opened. For each project only one session is possible and an administrator can only open one session at the same time. After an hour of inactivity the session can be closed by another administrator.

Object type

Object types are displayed in the protocol for a milestone in d.3 admin and mark the type of a master data object, e.g. a document type or a property field.

Action

Actions are displayed in the protocol for a milestone in d.3 admin and mark, if an object has been inserted (INSERT), updated (UPDATE) or deleted (DELETE).

Synchronization id

Every master data object gets a synchronization ID at the first start of d.3 admin after an update to the d.3 version 8, which is used as a reference for the transmission of transport data to other d.3 repositories. This provides the synchronism of the master data between the d.3 repositories.

Recording of the changes

Changes done in edit mode by the administrator are saved within a milestone and are therefore recorded for the transport into another d.3 repository. Not all changes can be recorded, displayed in the protocol and transported.

Practical hints

How can I work with projects and milestones, how can I start with it after an update?

The general work with projects and milestones is explained in the previous chapters.

If only one repository is used, after an update you create projects and you can make changes by using the edit mode.

If two repositories are used which have the same structure and kept synchronized, there are two options:

1. You build up a new test system based on the latest backup of the productive system
2. The plugin "d.3 project manager" adjusts the synchronization ID's of the master data objects. This only works reliable, if the structure of the two repositories are almost identical indeed (repository 1 contains everything from repository 2 and possibly some additional data). This assumes, that no projects are created yet. This step must be executed directly after the update.

If you are running several repositories with a different structure, you have to build up a new repository based on the data backup.

Which prerequisites must be fulfilled to use the transport system?

- The source system and the destination system must be on version 8
- The source system and the destination system contain the same master data objects
- The destination system must be build up from an export of the source system or
- The synchronization ID's of the master data objects of source and destination system are identical (using the plugun "d.3 project manager")
- The transportable data (see next point) are always transmitted from the source system to the destination system
- No changes on the master data objects are done in the destination system

What is covered by the transport system?

The following master data objects are considered by the transport system:

- Datasets
- Document types
- Advanced properties
- Multiple languages
- Dossier generation
- Groups (except memberships of users and groups)
- Document classes
- Authorization profiles
- Authorization profile assignments (not for users)
- d.3 organizational structure
- d.3 policy manager (not for users)
- d.3 restriction set manager (not for users)
- d.3 config (not for all parameters)
- d.ecs rendition service assistant

The following must be adjusted afterwards:

- User and group memberships, authorization profile-assignments, user-specific rights and properties as well as restriction set-assignments
- Hook-functions (Groovy-hooks can be transported from version 8.1.0 Rising 2, too)
- LDAP settings
- Repository dependent configuration parameter (e.g. hostimport paths)
- Plugin "d.3 explorer folder browser"
- Plugin "Automatic storage"
- Plugin "d.3 client config"
- Plugin "d.3 content service"
- Plugin "d.3 abo service"
- Plugin "d.3 search admin"
- Plugin "Form assignment"
- Workflows

> **Note**
> Changing user passwords can also be done in view mode!

Initial transport vs. subsequent transport

What is needed to be pointed out for the initial import? / What is needed to be pointed out for the export for the initial import?

The initial import, so the import into an empty repository can either be done by using a backup file (data backup) or by importing a project- or milestone. When exporting for an initial import you only

have to consider the following, if the source system was set up before version 8 and thus, administrative changes were done before the update: The milestone export contains only the changes on the master data which were explicitly executed. This also considers the dependencies to other objects. But other not dependent objects will not be considered. If you want to adopt these objects into the export, you either have to make some changes (by clicking through the objects) or you use d.3 project manager to assign the objects to the current milestone.

How do I handle parallel projects?

Using the same property fields (e.g. date or name)/using the same datasets (e.g. Yes/No)/What must be considered, if project 1, e.g. created the property but project 2 is set productive at first?

The export always considers the current state of the data and additionally holds the time of the last change on the object. This is considered during the import so that an older change does not overwrite a newer one.

What must be considered when removing document types, properties, etc.?

Nothing on principle.

If a document type is deleted in project 1 which was changed previously in project 2, the following would be done during the import:

If project 2 is imported before project 1, the document type will be initially created or changed. When importing project 1, the document type will be deleted.

If project 1 is imported before project 2, the document type will be deleted or not considered. When importing project 2, the document type will not be created.

How does the transport work technically? How does the import and export work?

Export

When closing a milestone the current build of the master data (e.g. a document type), which were changed in the edit mode of the milestone, are determined and written into a XML-file. All dependencies (e.g. the advanced properties of the document type) are considered and written into XML-file. Additionally the timestamp of the last change (independent, if the object was changed in the just closed milestone or in another) is written into the XML-file. This timestamp is then used for the import, so that older changes do not overwrite newer changes. The created XML-file is imported by the master async in the document type "Audit Log".

If a milestone is exported then, the XML-file will be downloaded and included in a ZIP-file. This ZIP-file contains additional control files.

You can export several milestones at the same time (Attention: The list of milestones can only be of one project and must be consistent). Also, a complete project can be exported, whereby only the closed milestone are considered.

Import

Before the import an initially consistency check of the data of the ZIP-file is performed. For each file a hash-value is calculated and adjusted with the hash-values from the .hash-file. A chnage of the files is not allowed to make secure that the import is performed on the rules of the transport system. If the .hash-files does not exist, no import can be executed.

During the import the objects are processed on their appearance in the XML-files. If master data already exist, then the timestamp of the last change in the destination system is compared with the timestamp of the last change from the source system in the XML-file. If the timestamp of the XML-file is younger than the one of the destination system, then the object with the data from the XML-file will be updated.

If there is an error during the import, the whole import process is cancelled and all changes are discharged.

What is the recommended procedure for several systems (test system, approval system, productive system)?

The initial point should always be a system (this means here the "test system"). Changes on the master data objects are done here. With help of the export from the test system the information are transmitted to approval system. Then, changes should be done on the test system. If everything is ok for an import to the productive system, this is done from the test system, but an export from the approval system would also be possible.

Installation of the basis workflows

The installation of the basis workflows can be executed as follows:

- Creating a project on the test system for the implementation of actions required by the basis workflow
- Enabling the d.3 parameter **WORKWLOW_SUPPORT**
- Configuration of the document class **FORWARDING** with the ID **ALL**
- Configuration of the user group **wfl_admin**
- Creating a user with the needed rights for the basis workflow installation
- Closing the milestone
- Installation of the basis workflows via the setup.

If you also want to install the basis workflow in the productive system, then use the export of the milestone which was created when configuring the master data objects on the test system and transport these into the productive system. The installation user is not considered by doing this and must be created manually on the productive system. Afterwards, the setup for the basis workflows can be executed for the productive system.

Changes on the destination system

You can still make changes on a productive system, i.e. the destination system of transport files. But you cannot change master data which are created by the transport system. However, changes on users and d.3 parameters also occur in the productive system. To perform those changes, you can create an "emergency project" and execute the changes within this project. Usually, you do not find any entries in the protocol because e.g. users or certain d.3 parameters cannot be transported and are thus, not recorded.

1.6. Retention periods and deletion of documents

This chapter gives you helpful information about retention periods and the deletion of documents.

Definitions

A document contains properties and document versions. A document version contains only useful files (the original application file and dependent files).

Retention period

The retention period describes the minimum storage time of a document in d.3ecm before it may be deleted.

It is configured in d.3 admin per document type in months. When a new document is imported, the expiry date of the term is determined and assigned to the document. If new versions of this document are created, the expiry date of the term for the document is also redetermined. In the case of transfer to a secondary storage facility, the expiry date is supplied.

If a "perpetual" retention period has been configured for a document type, no expiry date is stored for the document. However, when transferring to a secondary storage, an expiry date of "import plus 50 years" is passed in this case.

If event-oriented deletion has been configured for a document type, the retention period configured for this begins when an event is triggered later. This event is triggered via the d.3 server scripting api function **document_start_lifetime** per document. In this case, no expiry date is stored when a document is imported. An expiry date of "import plus retention period" is set when transferring to the secondary storage.

If the retention period is specified by a key date, a large number of documents can accumulate on one day when "deleting via d.3 async by moving them to the recycle bin" (see below). For more information on the processing speed of d.3 async, see the documentation d.3 batch.

1.6.1. Overview over the deletion of data

Basically d.3 admin allows the following settings for each document type:

- Retention time of the document in months
- Event-based deletion of a document or event-independent deletion
- Combination of time to live (>1 month lifespan) and event-based deletion

Additional information on the settings can be found in the chapter Document type | Retention time.

1.6.2. Static calculation of the retention time

If you configured a retention period and deactivate the event-triggered deletion, the import process statically calculates the "death date" and enters it in the d.3 database. The retention time is reset anew for every version of the document.

Example: An order confirmation that is imported on 12/11/2016 leads to the static determination of the end of the retention period 12/11/2022, taking into account the set lifespan of 72 months.

If the additional parameter **End <month name>** was set, the end of the retention period is set to the last day of the month.

Please note that a great number of documents get the same retention time this way which can result in significant performance issues.

> **Warning**
> To avoid problems, it should ideally be avoided to switch between static and event based calculation of the retention time.
>
> Such a step usually requires subsequent corrections of the existing data which is done using (customer-specific) scripts (e.g. **korrigiere_sterbedatum.jpl**, **retention2storage.jpl**). These are provided by the d.velop AG on request.
>
> Please note that such a retroactive correction is no longer possible under certain conditions!

1.6.3. Dynamic calculation of the lifetime

If you want to control dynamically (event-based) how the retention time of a document is calculated and entered in the database or the secondary storage system, then enable the additional option **Event-based deletion**.

When configuring the retention period of a document and activated option for the event-triggered deletion in d.3 admin no retention time is written to the data base, i.e. the document waits for a start event to begin the defined time span.

If this event occurs, then the specified retention period in months is added to the current date and this created date is written to the database as retention time.

1.6.4. Defining the event

For the dynamic calculation of the retention period, an event must first be defined which should lead to the dynamic calculation of this retention period. This event can be, for example, a product discontinuation or a definition of discontinued articles. This event could for example be triggered by a status transfer in the product dossier or the respective document type and would lead to the setting of the retention time in the d.3 tables through a hook function.

> **Note**
> In order to make use of the event-based deletion, you should call the Server API function **document_start_lifetime** in a hook function.

1.6.5. Deleting documents

On the following pages you will find out which ways there are to delete documents and which instructions are to be observed.

Deleting via API

Please note the following notes when deleting documents by moving them to the recycle bin if you use the d.3 api function **DeleteDocument**:

1. After deletion, data remains in the recycle bin.
2. All versions of a document with the status Processing, Verification and Release must be deleted one after the other (that is, individually). To delete all document versions, **DeleteDocument** may have to be called several times (up to three times).
3. The archive versions of a document can only be deleted when no other versions from 2 are available.
4. Unlike 2, all archive versions are deleted with a single function call.

Deleting documents in the status Processing or Verification

a) There are more document versions in Release or Archive.

When deleting, the document version is moved to the recycle bin, but the document remains searchable.

Nevertheless, the document version cannot be restored later (exception: the following parameters were set: **DELETE_FILES_FROM_PROCESSING** or **DELETE_FILES_FROM_VERIFICATION**).

b) There are no more document versions in Release or Archive.

When deleting, the document version is moved to the recycle bin and is no longer searchable.

Nevertheless, the document version cannot be restored later (exception: the following parameters were set: **DELETE_FILES_FROM_PROCESSING** or **DELETE_FILES_FROM_VERIFICATION**).

Deleting documents in the status Release

For deleting, enabling the global parameter **ALLOW_DELETE_FROM_RELEASE** in d.3 admin must be done first. Afterwards, additional user rights are available in d.3 admin, which must then be added to the user.

a) There are more document versions in the status Archive.

When deleting, the document version is moved to the recycle bin, but the document remains searchable.

The document version can be recovered from the recycle bin.

b) There are no more document versions in the status Archive.

When deleting, the document version is moved to the recycle bin and is no longer searchable.

The document version can be recovered from the recycle bin.

Deleting documents in the status Archive

When deleting, the document version is moved to the recycle bin and is no longer searchable.

The document version can be recovered from the recycle bin.

Please note the following when deleting documents by moving them to the recycle bin if you use the function **document_delete** from the d.3 server scripting api (jpl + groovy):

The behavior here is the same as for the API function **DeleteDocument** with the following exceptions:

1. The parameter **ALLOW_DELETE_FROM_RELEASE** is only relevant if the parameter **user_name** is set.
2. If the parameter **del_from_each_status** is set to 1, then the document will be moved with all document versions to the recycle bin.
   The document version can be recovered from the recycle bin.

Deleting privileged via API

Please note the following notes for using the API function **DeleteDocument** from the d.3 api or **document_delete** from the d.3 server scripting api:

1. After deletion, all data of the document are removed from d.3ecm. Only a minimal entry remains in the database:
   - Document ID
   - Document type
   - Document number
   - Deletion data
   - d.3 user who has deleted the document on a privileged basis
   - User-defined reason for deletion (free text).
2. The privileged deletion is only possible for documents that are searchable in d.3ecm. Documents, which are in the recycle bin with all versions, privileged deletion is currently **not** possible.
   These documents can also be completely deleted via a minimum standard retention period of documents in the recycle bin.
3. A special license key is required in the customer license.
   If this exists, an additional user right for privileged deletion can be assigned.
4. If document versions in status Release or Archive are to be deleted, the parameter **ALLOW_DELETE_FROM_RELEASE** must also be set in d.3 admin.
5. All versions of a document with the status Processing, Verification and Release must be deleted one after the other (that is, individually). To delete all document versions, **DeleteDocument** may have to be called several times (up to three times).
6. The archive versions of a document can only be deleted when no other versions from 2 are available.
7. Unlike 2, all archive versions are deleted with a single function call.
8. If secondary storage is available:
   The following storage systems support privileged deletion before the retention period expires. Please note that this information is developer-specific and must be checked by you in each individual case. This information is supplied without liability:
   - • NAS
     - Netapp

- TSM
- Centera
- HCP
- iCAS

Deleting via d.3 async

The deletion of documents, on which the retention period is exceeded, is performed by d.3 async. The delete function is disabled by default and must be enabled in d.3 admin under **d.3 config > d.3 asynchronous processing**.

Deleting documents after the retention period has expired

Please note the following notes when deleting documents after the retention period has expired by moving them to the recycle bin via d.3 async:

1. After deletion, data remains in the recycle bin.
2. If a retention period is configured for a document type in d.3 admin and the parameter **DELETE_DOCS_ELAPSED_AUTO** is set, documents are automatically moved to the recycle bin by d.3 async after this period has expired.
3. The document with all its versions is moved to the recycle bin.
   The document version can be recovered from the recycle bin.
4. In order to prevent a permanent load for the d.3 async process, the maximum number of documents to be deleted per job can be limited with the parameter **DELETE_DOCS_ELAPSED_PER_CALL = "50"**. d.3 async checks every 15 minutes if there are documents to be deleted. The specified number of documents is deleted per deletion process.

Deleting documents by "event-oriented deletion"

Please note the following instructions when deleting documents by "event-oriented deletion" via d.3 async by moving them to the recycle bin:

1. After deletion, data remains in the recycle bin.
2. If event-oriented deletion has been configured for a document type, the retention period configured for this begins when an event is triggered later. This event is triggered via the d.3 server scripting API (JPL und Groovy) function **document_start_lifetime** per document.
3. The procedure described above "Deleting documents after the retention period has expired" then takes effect.

Deleting documents via d.3 async by removing them from the recycle bin

Please note the following instructions when deleting documents via d.3 async by removing them from the recycle bin:

1. After deletion, all data of the document are removed from d.3ecm. Only a minimal entry remains in the database:
   - Document ID
   - Document type
   - Document number
   - Deletion data
   - d.3 user who has deleted the document
   - User-defined reason for deletion (free text).
2. The default retention period of documents in the recycle bin is 365 days by default and can be adjusted in d.3 admin (parameter **RECYCLE_STORAGE_PERIOD**)
3. If the parameter **DELETE_DOCS_ELAPSED_PHYSICAL** is enabled in d.3 admin, d.3 async will automatically delete all documents from the recycle bin whose retention period in the recycle bin is longer than the entered value in **RECYCLE_STORAGE_PERIOD**.

Deleting documents from secondary storage

With the parameter **DELETE_DOCS_ELAPSED_SEC_STORAGE = "1"** d.ecs storage manager can be instructed to also delete the documents from the secondary storage.

> **Warning**
> If files of a document have been moved to a secondary storage, a deletion request is sent to d.ecs storage manager. The document files are only removed from the secondary storage when the retention period for the document originally transferred to the secondary storage expires. There may be years between the deletion request and the expiry of the retention period.

Deleting via script

Apart from the deletion with the d.3 async process the deletion can be performed with a delete script running simultaneous to the d.3 async process. This alternative should be preferred with a high volume of documents to be deleted.

In this case, please contact the d.velop AG. The use of delete scripts is always a customer and project-specific action and the defined deletion scenario must be negotiated with the customer and properly documented.

Examples

Example 1

The lifetime is set in the document type as 72 months, event-based deletion is not enabled.

If the last change of the document lies more than 6 years in the past, then the document is moved to the recycle bin by d.3 async.

Example 2

The lifetime is set in the document type as 72 months, event-based deletion is enabled and a JPL script is created handling the according event.

If the event occurs, e.g. the product cancellation, then the retention time is set to "today" plus 72 months via the script and the document is thus deleted 6 years via d.3 async.

Retention time on the storage:

An event-based document is written to the storage system (e.g. NetApp) with the defined retention period (e.g. 72 months). If the d.ecs storage manager has not received any information on the document shortly before the end of the retention period because the event has not yet occurred d.3, then the document lifetime is extended by at least 180 days until the event occurs.

Afterwards, the d.ecs storage manager receives a retention-job by d.3, which irrevocably sets the document's retention period on the NetApp.

1.7. Troubleshooting

1.7.1. Advanced properties fields

Change numeric data type to alphanumeric

> **Warning**
> Before any change to the system we urgently recommend a backup of the data.

Find out before this administrative operation, in which document types the (numeric) advanced property field is used. For this effect, open d.3 admin in the overview of the advanced properties fields and select the desired property field of the document type with a right-click, and open the context menu option **DB Fields Overview**. Here you can sort against the column Title.

If the advanced property field is used in multiple document types, then you can select the section and save it externally using the context menu option Save selection as .... Afterwards, you can change the database-position via the properties of the advanced properties field. To do so, select a database-position between 1 and 49.

Once the advanced property field was modified regarding the database-position, you must also change the document types containing this advanced property field and adjust the database-position.

> **Warning**
> Check before the change of the database position, if it is possibly in use by another advanced properties field in one of the document types.

Having applied these administrative changes all newly imported documents are now stored to an alphanumeric database-position.

For the existing documents in the repository, the content of the 80-field must be copied to the modified alphanumeric database-field as the advanced properties field-content would not be displayed and searched.

> **Note**
> If an existing folder scheme uses the advanced properties field as a recognition property, then this must also be adjusted.
>
> A retroactive change of the data type of a property is only possible if the property is not used in a document type!

1.7.2. Managing users

Passwords not compliant with the rules

Validate in which way the password for the user was created. If the password was created in d.3 admin, then the security settings regarding the syntactic rules do not apply. These security settings regarding the syntactic rules only apply, if the password is changed via d.3 login manager (d.3 login). The user d3_admin can still assign the passwords in d.3 admin disregarding the specified syntax definition.

Deleting users in the administration

The following conditions must be met to delete a user:

1. 1. Emptying the mailbox: Clear sent and received objects (via **Remove / Acknowledge**)
2. 2. Clear all documents in the **Processing** or **Verification** of the user to be deleted.
3. There must not be any document in the d.3 document-tree under: **\bearbeit\user name**.

If the user cannot be deleted despite the above mentioned conditions, please check, if Microsoft Windows created a directory named **Thumbs** in the Processing-directory. With the Microsoft Windows-default settings this is not displayed (option **Hide protected operating system files** in the **folder options**).

Deleting or blocking users

You can block individual user in d.3 admin with a right-click and choosing the respective option. In order for this menu option to be visible, the following parameters must be set in d.3 config:

Password settings

Count failed logins

PASSWORD_USER_FAIL_CHECK = "1"

and

Failed login threshold

PASSWORD_INPUT_TRIALS = <number of failed attempts e.g. "3">

OR

Extended syntax for passwords

PASSWORD_EXTENDED_SYNTAX = "Yes"

This settings results in the highest level of security for the password. Only passwords complying with the configured syntax rule may be used:

At least one character from at least three of the following four groups:

- A-Z
- a-z
- 0-9
- Special characters

The password expires after the specified number of days. None of the n recently used passwords may be used whereby the value for "n" is configured with the parameter **PASSWORD_HISTORY_ENTRIES**.

These password-settings only affect changes via the d.3 login manager. The user account is blocked after several failed login attempts. Having enabled extended syntax, all users must change the password.

1.7.3. Mailbox

Configuring SMTP-mail for the resubmission

1. Start d.3 admin.
2. Open the configuration (**d.3 config**) under the **System settings**.
3. Make the following the settings in the section **d.3 SMTP support**:
   a. **Enable SMTP support**
   b. **Enable SMTP for the resubmission**
   c. **Attach document reference to mail**
   d. **SMTP gateway**
   e. **SMTP subject line**
4. Confirm these configuration.

> **Note**
> Check if every user in d.3 admin has an e-mail address assigned in the user management. With these settings, the user should now be notified by e-mail on receiving a resubmission. If this should not work as expected you should check if it is a configuration problem on the e-mail server.
>
> To do so, use the command line test program **sendmail.exe** to check if sending SMTP-mails works.

The required command line is this:

```
sendmail.exe Sender Empfänger Textdatei [Betreff]
```

In the **sendmail.ini**, the mail server must be specified.

```
[SendMail]
Server=<SMTP-Servername oder IP-Adresse>
```

The test program **sendmail.exe** is located in the directory **.\d3\d3server.prg**.

Using the refining search in the d.3 smart explorer mailbox

If you want to use the function for the refining search in the d.3 smart explorer in the mailbox as well, this can be done by configuring a dummy document type.

The thus configured dummy document type is then also available in the mailbox with its properties for the refining search.

Additional information on the detailed search can be found in the manual for d.3 smart explorer.

1.7.4. Edit Document Types

Meaning of the document type audit log

> **Warning**
> With d.3 version 7, the logging of the administrative changes is no longer written to this document type! Administrative modifications will be made in an audit-trail and therefore directly written into the database. Initially, this information can only be accessed via d.3 API calls (GetAuditData).

All information is stored in the database tables:

audit_trail, audit_trail_action and audit_trail_detail.

The layout of the tables looks as follows:

Table audit_trail	
Column	**Meaning**
audit_id	Primary key and the clip around all entries to an API call, an async job or a hostimport action
app_id	Three-digit code of the calling application (d.3 product ID from the numberband)
module_id	The three-digit module id/version ID that is sent from the calling application
timestamp	Time at which the operation is performed
editor	The d.3 user ID of the caller (this can also be „d.3“ itself) This is the actor/the subject of an action, that is the one that performs the action
editor_host	IP address from which the API call came; and for a multilayer architecture (e.g., with d.3one) this is not necessarily the computer to which the user is actually working.
srv_status	Internal administrative data for technical reproducibility and debugging purposes (Format: "ProcessID-ApiCallNumber", with this an audit trail entry can easily assign to an entry in the technical d.3 log).

Table audit_trail_action	
Column	**Meaning**
audit_id	Foreign key to the table audit_trail
action_id	Together with the column audit_id it builds the pimary key of this table. If within one API call several objects are processed or more actions are performed on an object, these actions can be distinguished by the action_id.
object_type	The type of the object in d.3 which is processed/edited
object_id	Unique identifier of the object on which the current action is executed

Table audit_trail_action	
action	Predefined identifier of the type of action that was executed by the editor on the object (create, edit, delete, ...) (for details see the d.3 API documentation).

Table audit_trail_detail	
Column	**Meaning**
audit_id	First part of the foreign key to the table audit_trail_action
action_id	Second part of the foreign key to the table audit_trail_action
attribute_name	Name of changed object property; the values for this field in general agree with the corresponding identifiers in the d.3 API
attribute_value	The new value of changed object attribute

For further information please see the documentation of d.3 server API.

With d.3 version 7, the document type AuditLog is only used for d.3 configuration changes, i.e. changes to the files:

d3addon.ini, d3config.ini, d3admin.ini, AdminImportLog, AdminLog.

Using title fields

Title fields are used to improve the display of document properties in d.3 smart explorer.

In older d.3 versions, in the d.3 smart explorer link-view the document number was displayed together with the repository ID of the d.3 repository and the document type such as D000065565 invoice. Since this did not seem to be meaningful enough, the possibility was created to individually design the document recognition by defining the title fields per document type.

Title fields can be defined as follows:

In d.3 admin, you can assign each document- or dossier type with an individual title. This setting is located in the Document types.

This title definition only applies to the dossier and document types in which the definitions are made.

In the **Position window of Titels** you can select the relevant properties to be included in the title definition. To do so, select the corresponding property and double-click or click on **Properties**. Then you can make the appropriate definitions on the page **Title properties**.

If you define several properties for title fields, the order in the document type determines the structure of the overall title.

Example: If you want to assign more descriptive names to an order, you can define a title field in the document type "Order". To do so select under **Default** the field **Title** and set it to "Yes". Under **Title** you can define advanced properties:

Here you can set a suffix and a prefix and moreover limit the length of the displayed text in d.3 smart explorer (from - to).

Besides the links, the title fields are also used in functions such as **Senden to via e-mail**. The title definition and the file extension, e.g. doc is used to form the file name.

In the file **d3config.ini** you can create a formatting for a date field so that a date field can be used in sorting.

1.7.5. Example for restriction sets

Document classes via restriction sets

When defining the restrictions, you have several options.

- Constant expression, also with wildcards (*, ?, more are possible but must be defined)
- Macros: @D3USER, @D3GROUP, @D3REGEXPR, ...
- Restriction sets referenced with the macro @D3SET

Restriction sets are defined with individual values but also with data-ranges and negative expressions and are stored as a text document.

The sets are associated to the different target objects (user, group, global, ...) too. The set of values to be checked is then only determined at runtime.

Initially you create the "containers" for the restriction sets in the d.3 admin plugin d.3 restriction sets.

In the second step, you list the values against which the comparison is to be applied later as part of the validation of rights, i.e. the valid and prohibited values.

You can associate global values (no association), user- or group-related values, but also values associated to an organizational unit (company structure).

You best create the basic entries first (no association).

The, get more precise, e.g. add further entries for a group, where only the group members (Accounting) should get additional rights. If the name of the restriction set is the same as the name of a static dataset, then you can select from the configured values.

It is thus possible, to define basic values for all (group-)members and still to configure additional extensions and restrictions for a user or group.

The association is depicted accordingly.

You can view the associated values from a user or group perspective.

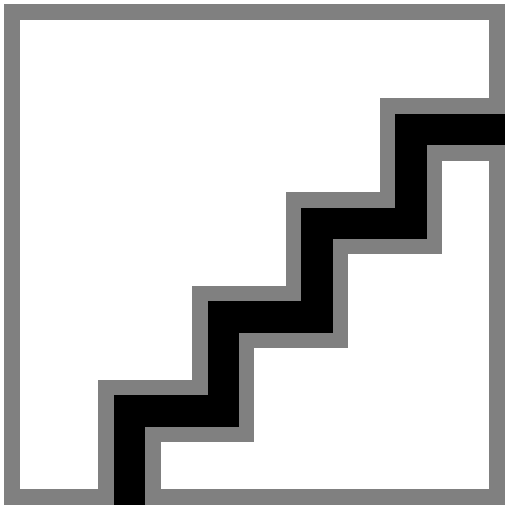
In the definition of the document class, you then use the macro @D3SET, ideally by clicking the button ... at the end of the property field to select the suitable restriction set.

This document class does not need to be adjusted any more, if you want to assign extended or reduced values to the users. It is sufficient to edit the values accordingly in the restriction set!

In order not to leave this task to the d.3 administrator, you can name users in charge of assigning the values in the restriction set with reference to their respective environment and thus to respond to the immediate requirements such as staff member on sick-leave.

For this effect, such a specialist in charge get the respective rights for such a restriction set via a document class for the system document type **$Restriction Set$**.

In this example, a document class is defined, where a person in charge is made eligible to only care for restriction sets with the name "Sites" and only those associated to users (1). The restriction sets are accessed via d.3 smart explorer and a respective entry in the Tools menu.



1.7.6. d.3 repository configuration

Changing the prefix of the document ID in the d.3 repository

The prefix of the document IDs in the d.3 system is to be modified, as the maximum number of document IDs for this prefix is reached.

To change the prefix e.g. from CA to C, you must open the **d3addon.ini** for the respective repository (...\d3\d3server.prg\d.3 archive name\d3addon.ini), where d.3 repository name stands for the respective d.3 repository such as d3R.

Append the file with an entry server = "x", where "x" stands for the desired prefix. Then, restart the hostimp process, the d.3 async server processes and the d.3 async process. All new documents receive the document ID as a number build from Prefix + Number, e.g. C0000001.

1.7.7. Server processes

If you want to specialize server processes for certain tasks, you have to configure an alternative d3fc server ID for the communication with d.3 gateway.

This is how it works

1. Create a process-specific ini file in the configuration directory of the repository.
2. In this file you assign the server ID by configuration variable **d3fc_alternative_server_id**.
   Example:
   Create the file **dxchange_api_import.ini** with the following content:

   ```
   d3fc_alternative_server_id = "E"
   ```

   This file is specified as process ini command line parameter for the added server processes. See parameters of **d3odbc32.exe**.
3. Now the processes with the new server ID can be targeted by the also for this configured applications.

1.8. Infrastructure components

1.8.1. d.ecs task

In order to use d.ecs task, you must first install it and register it with the d.ecs http gateway app.

Enable the option **ENABLE_TASKAPP** in the file **d3addon.ini** file to turn on d.ecs task support for the d.3 server.

> **Note**
> Also read the prerequisites and important information about enabling the **ENABLE_TASKAPP** option in our KB article 000001788 "How do I migrate my mailbox to d.ecs task with the ENABLE_TASKAPP parameter?".
>
> If errors occurred during migration after enabling the option, the KB article 000001797 "Fixing migration errors after enabling the ENABLE_TASKAPP parameter" may help you.

In future versions, d.ecs task will completely replace the d.3 mailbox. In one of the next d.3ecm current versions, this option will be enabled automatically for new installations accordingly. For update installations, the current status for using the d.3 mailbox or d.ecs task remains unchanged.

In d.3 config, enter **DECS_SERVICE_USER** and **DECS_SERVICE_USER_PWD** so that the d.3 server can authenticate itself to the Task app. Give the service user you specified in d.3 config the "Service user" role in the Task app. The setup for this can be found in the chapter "Setting up user roles" in the d.ecs task administration manual.

Enabling this has the following effects:

- All new mailbox entries are no longer created in the d.3 database, table wieder_vorlage, but are forwarded directly to d.ecs task.
- All API function calls with a direct mailbox reference (except **SendHoldFile**) run into an error.
- The previous mailbox entries are forwarded to d.ecs task and deleted from the table wieder_vorlage. Only the entries that have not yet been acknowledged are taken into account. All acknowledged entries are deleted.
- An entry is created in the history table with the action type WVMIG1 for all mailbox entries which have not been acknowledged.

A subsequent disabling of the parameter has the following effects:

- The entries at d.ecs task are not transferred back into the d.3 mailbox.

1.9. Additional information sources and imprint

If you want to deepen your knowledge of d.velop software, visit the d.velop academy digital learning platform at https://dvelopacademy.keelearning.de/.

Our E-learning modules let you develop a more in-depth knowledge and specialist expertise at your own speed. A huge number of E-learning modules are free for you to access without registering beforehand.

Visit our Knowledge Base on the d.velop service portal. In the Knowledge Base, you can find all our latest solutions, answers to frequently asked questions and how-to topics for specific tasks. You can find the Knowledge Base at the following address: https://kb.d-velop.de/

Find the central imprint at https://www.d-velop.com/imprint.