Administrator Manual

requisis_**Toolbox**

Instructions for the Installation and Configuration of requisis_Toolbox Version 4.7

requisis_Toolbox

Table of Contents

requisis_Toolbox

1 Introduction

requisis_ReqIF-Manager is part of the requisis_Toolbox for DOORS Next Generation (DNG) to exchange modules using the Requirements Interchange Format (ReqIF respectively RIF) between different requirement management databases and your **IBM DOORS Next Generation (DNG)** database. This document provides detailed information for administrators and experienced users about the installation and configuration of the tool.

2 Requirements

To run the *requisis_Toolbox* the following conditions are required:

- IBM DOORS Next Generation 6.0.6 or later
- IBM DOORS Next Generation 7.1 is supported from Version 4.7
- Local Installation of Docker Engine or Docker on a server and Docker compose
- Delivery Package of *requisis_ ReqIF-Manager* provided by REQUISIS
- We recommend allocating at least 4GB of memory each to the *Toolbox Server* and *mySQL* containers at the start.

IMPORTANT: If you receive error messages such as E501: The job terminated unexpectedly

Increase the memory in your configuration. The number of parallel jobs, the number of users and the workload of the DNG server and the toolbox itself have an influence on the required memory.

3 Installation

To use the *requisis_Toolbox* and its individual components, you need to install the Docker Desktop and Docker Compose on your machine.

The installation of *requisis_ReqIF-Manager* is divided into the following steps:

- Get the *requisis_Toolbox* zip-package from REQUISIS
- Configuration of an oAuth-Consumer in JTS
- Configuration of various settings like DNG Server Configuration, LDAP configuration
- Creation of SSL certficates and placement in docker storage folder
- Starting the Docker Container

3.1 Download the requisis_Toolbox with the requisis_ReqIF Manger

Download the zip file from the REQUISIS download portal (https://requisis.com/de/apps/file-download) with your personal download code and unzip it. If you need a new personal download code, please contact the product support at product-support@requisis.com .

In case this is an Update please unzip the package to a new location first and update your existing files as described in chapter 8 Update procedure for new release.

3.2 Description of Package and Container

3.2.1 Required Containers

Mysql

The mysql-container offers the database services for the requisis_Toolbox.

The container is downloaded from dockerhub.

Requisis-toolbox

The requisis-toolbox container offers the functionality of requisis_Toolbox.

The container will be downloaded from a private repository of requisis. In order to download the container, you need a login that will be provided by REQUISIS as long as you have a valid support contract.

3.2.2 Mounts

The Mount Point column contains the path within the corresponding Docker container. In the Mount target column, you will find the path after successful installation from your installation package in the file system.

mysql

Mount-Point in Container	Mount target	Description
/docker-entrypoint-initdb.d	./dist/conf/mysql/sql	Initial SQL-DB content
/var/lib/mysql	./store/mysql/data	MySQL Database content
/var/log/mysql	./store/mysql/log	MySQL Logfiles

toolbox-server

Mount-Point in Container	Mount target	Description
/store	./store/toolbox/config	Toolbox Server configuration
/app/web/uploads	./store/toolbox/uploads	Content that has been uploaded by user and is being processed by requisis_Toolbox
/app/var/logs	./store/toolbox/logs	Logfiles of requisis_Toolbox

3.2.3 File and Folder Structure of deployment package

Path and File	Description
docker-compose.yml	Docker-compose file. Used to bring up all relevant docker containers
.env	Contains Names/URIs for Docker Images and mysql password.
startDocker.cmd	Windows Shell Script to pull the latest images and start docker container.
createMysqlBackup.cmd	Windows Shell Script to create a mysql database dump.
store\mysql\data	Will contain the content of the mysql database. Is empty when delivered.
store\mysql\log	Storage of mysql logs. Is empty by default.
store\toolbox\config	Contains the configuration of the requisis_Toolbox
store\toolbox\config\parameters.yml	Contains the configuration parameters of requisis_Toobox
store\toolbox\config\mix-config.yml	Contains the configuration of DNG-Servers. (Configuration of the requisis_MiX library which is used by requisis_Toolbox to communicate with DNG)
store\toolbox\config\features.yml	Contains the licensed and enabled features. **This file will be provided by REQUISIS and must not me modified.**
store\toolbox\config\routing.yml	Contains URL path parameters. **This file will be provided by REQUISIS and must not me modified.**
store\toolbox\config\.init	This file indicates that the store has been initialized. **This file will be provided by REQUISIS and must not me modified.**
store\toolbox\config\mix-certs	This folder contains certificates that might be needed to access the DNG Server.
store\toolbox\config\ssl	This folder contains the ssl certificates that are used by the webserver.
store\toolbox\config\ssl\server.crt	Contains the server certificate.
store\toolbox\config\ssl\server.key	Contains the server certificate private key. **The key file must not be password protected!**
store\toolbox\logs	Contains the logs of the requisis_Toolbox

Path and File	Description
store\toolbox\uploads	Contains all the data that is stored by the requisis_Toolbox (besides the data that is stored in the database)

3.3 Setup initial package

After unpacking, copy the initial folder provided by requisis to the desired location in your system and perform the following steps:

> Copy the **features.yml** file from the email to the subfolder *store/toolbox/config*
> Open the **.env** file with an editor at the top level of the package and in the *HARBOR_USER=* line add the harbor user you provided and, in the *PW=* line add the password you have received.

4 Configuration

Before the *requisis_Toolbox* and the *requisis_ReqIF-Manager* can be started, you must make several settings according to your infrastructure on a few files. You also need to access the DNG server configuration.

4.1 Configuration of DOORS Next Generation

IMPORTANT: To make *requisis_ReqIF-Manager* working with DNG you need to setup an OAuth Consumer Key and Secret in the JTS Application of the Jazz Platform.

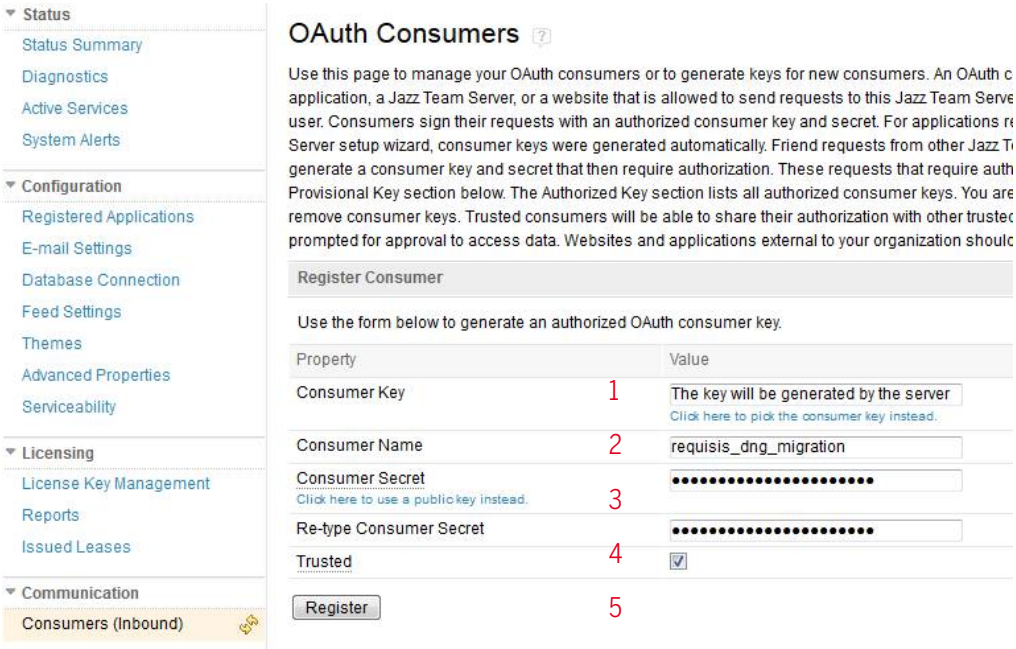
Do not setup the oAuth Consumer Key and Secret in the RM Application!

The Consumer Secret must consist only of capital and small letters and numbers.

Instructions:

Open "**https://(your-server:your-port)/jts/admin#action=com.ibm.team.repository.admin.configureOAuth**".

The following page should open:



1) Set *Consumer Key*: Either choose one or note down the key generated by the server.
2) Set *Consumer Name*: Set to "requisis_ReqIF Manger" (or whatever you like)
3) Set *Secret Key*: Choose a Secret from upper- and lower-case letters and numbers. Special characters are not supported!
4) Set *Trusted*: Activate this checkbox
5) Click on *Register*

4.2 Configuring DNG-Servers in mix-config.json file

Location: store\toolbox\config

Important Note: You need to restart the docker container to allow the docker container to pull the new configuration.

Name in Jazz Platform	Parameter in configuration.json
Consumer Key	`oauth_clientId`
Secret Key	`oauth_clientSecret`

Place your DNG server URL in the *targetUrl* line. You can configure multiple servers. The file should then look like the image below. In this example two servers were configured.

```
{
  "serverTargets": [
    {
      "targetUrl": "https:\/\/your-server1:9443",
      "oauth_autoLogin": true,
      "oauth_clientId": "tbd",
      "oauth_clientSecret": "tbd",
      "maxSynchronousCurlRequests": 5,
      "databaseCreatedPriorToVersion7": true
    },
    {
      "targetUrl": "https:\/\/your-server2:9443",
      "oauth_autoLogin": true,
      "oauth_clientId": "tbd",
      "oauth_clientSecret": "tbd",
      "maxSynchronousCurlRequests": 10,
      "databaseCreatedPriorToVersion7": false
    }
  ],
  "maxSynchronousCurlRequestsDefault": 5,
  "proxy": "",
  "indexerCheckInterval": 100,
  "preferPublicApi": true,
  "curlTimeout": 120,
  "curlConnectionTimeout": 60
}
```

Please note that slash `/` needs to be escaped with a backslash `\` in the json format and don't forget to save the file for your changes to take effect. For DNG databases created in version 7 or later, no Correlator service is required. With the parameter `databaseCreatedPriorToVersion7` you enable the Correlator service for your databases for version 6 or older. Set the parameter to *false* if the database used was created with version 7 or later.

After the serverTargets section are the following options that apply to any connection to the above targets.

- `maxSynchronousCurlRequestsDefault:`
  Number of synchronously executed requests to the DNG server. Tweek for performance optimization.
  Default value: 5
  Only change if advised by REQUISIS to do so.
- `proxy:`
  If necessary, an IP address and port number (syntax: host:port) for a proxy server can be entered here for the connections to the JAZZ server.
  Default value: (empty)
- `indexerCheckInterval:`
  Interval in seconds while waiting for tasks on the JAZZ server
  Default value: 100
  Only change if advised by REQUISIS to do so.
- `preferPublicApi:`
  Prefers to use the DNG public API before the private API
  Default value: true
  Only change if advised by REQUISIS to do so.
- `curlTimeout:`
  Timeout while waiting for a response from the JAZZ server in seconds.
  Default value: 120
  If the DNG server is busy, setting this parameter high can prevent timeouts in the ReqIF-Manager.
- `curlConnectionTimeout:`
  Timeout while waiting for the connection to the JAZZ server in seconds.
  Default value: 60
  If the DNG server is busy, setting this parameter high can prevent timeouts in the ReqIF-Manager.

4.2.1 SSL-client certificates for DNG Server Connections (optional)

This configuration step is only necessary if *SSL client certificates* are required for communication with the DNG server. To set up SSL client certificates for a specific server you need to edit the *mix-config.json* file.

In the relevant section of the server concerned, insert the lines `ssl_client_cert_file`, `ssl_client_cert_keyfile` and `ssl_client_cert_keyfile_password`.

As shown here:

```
{
      "targetUrl": "https:\/\/your-server1:9443",
      "oauth_autoLogin": true,
      "oauth_clientId": "123ab12a",
      "oauth_clientSecret": "123ab12a",
      "ssl_client_cert_file": "conf/certs/ssl-client-cert.pem",
      "ssl_client_cert_keyfile": "conf/certs/ssl-client-cert.key.pem",
      "ssl_client_cert_keyfile_password": "yourSecretPassword",
      "maxSynchronousCurlRequests": 5
},
```

Put your SSL public and private certificates for the connection to your DNG server here:

`store\toolbox\config\mix-certs`

The certificates will then be copied to the docker container during the next start of the docker container.

4.2.2 Warning: There are more DNG servers configured than licensed.

If this warning is permanently displayed at the bottom of the screen, the DNG servers entered in the *mix-config.json* file must be adjusted. There are more DNG servers configured in the file than licensed.

Contact requisis Product Support if all configured servers are needed to adjust your license accordingly. Otherwise remove DNG servers from the file that are no longer used.

4.3 Settings in parameters.yml

Location: store\toolbox\config

Important Notes:

You need to restart the Docker Container to allow the docker container to pull the new configuration.

Make sure that no parameter is duplicated in the file, otherwise the Docker Container will not start up!

Some parameter names and syntax have been changed from Version 1.8 to 1.9. Please also refer to the chapter No longer used parameters

The file is divided into several sections by comments. The headings in the file can be found here as sub-chapters.

- `database_host`:
  Host address of your Database. Default value: mysql if the MySQL instance in the docker_compose.yml is also named like this.
- `database_port`:
  Port of your used database. Default value: 3306
- `database_name`:
  Database mysql table for working. Default value: requisis_toolbox
- `database_user`:
  Username for the database login. Default value: toolbox
- `database_password`:
  Password for the username to login in your database. Default value: mysql
- `app_url: 'http://myELM.com/requisis-toolbox'`
  Base URI and context root under which the Toolbox is accessible. The app_url is required to enable the full functionality of the Toolbox. It must include everything from the Toolbox URL that comes before /frontend (without a trailing slash).
- `secret:`
  Salt for encryption / decryption, do not change after installation
- `logger.name`
  Name of logfile. Default value: 'requisis_toolbox.log'
- `logger.level`
  loglevel for logging possible values are: debug, error, warning, critical, info Default value: 'error'
- `logger.dng_communication_log.log_always: 'true'`
  writing the DNG communication log without debug mode
- `directory.logs`
  webserver path to logfiles. Default value: ./web/logs
- `directory.files.temp:`
  webserver path for temporary file uploads of requisis_Toolbox. Default value: './web/uploads/temp/[USER]/'
- `directory.files.limit.uploaded`:
  maximum file uploads for requisis_Toolbox. Default value: 20
- `directory.files.user_uploaded:`
  webserver path for file uploads of requisis_Toolbox. Default value: './web/uploads/uploaded/[USER]/'
- `directory.files.user_storage:`
  webserver path for stored files of requisis_Toolbox. Default value: '/web/uploads/storage/[USER]/'

- `auth.session.timeout:`
  number of seconds of inactivity after which the session becomes invalid. Default value: 3600

4.3.1 LDAP Login

Important Note: All values from the LDAP/Actice Directory must be transferred **CaseSensitive**. The keywords are only entered in **capital letters**. Ex. DC=lovelySupport,DC=HeLp

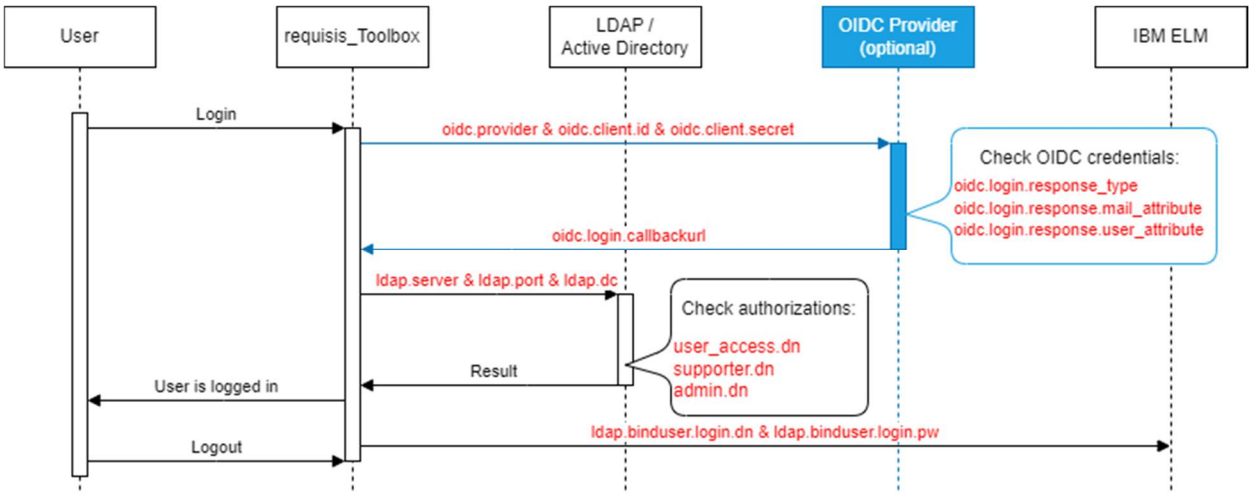
If you want to fargo LDAP and authenticate your users exclusively via OIDC, please refer to chapter 4.3.3 OIDC Settings and the subchapter OIDC login.

- `auth.ldap.debug: 'false'`
  If the value is set to 'true', the LDAP debug mode and the login.log are activated for more detailed error messages. This helps you to analyze problems with the login. You can also find the requisis_toolbox_login.log in the subfolder store/toolbox/logs
- `auth.ldap.server: 'ldap-server.company.net'`
  The address of your LDAP server / Active Directory.
- `auth.ldap.port: 389`
  The standard port for LDAP is 389 and 636 for secure connections.
- `auth.ldap.encryption: none`
  Enter the type of encryption used for ldps here. (*tls* or *ssl*) If you use ldap, enter *none* here.
- `auth.ldap.dc: 'DC=carcompany,DC=com'`
  LDAP path to the corresponding root or directory in which your organization has the relevant user data for the planned use of the requisis_toolbox.
- `auth.ldap.disabled.users.execute.command: true`
  Specifies whether the cron job cleanup users should be executed daily. You can find more information on this in the chapter User Cleanup Script.
- `auth.ldap.admin.dn: 'CN=ReqTbAdmin,OU=Admins,DC=carcompany,DC=com'`
  Users who are members of this group receive the admin role for the requisis_Toolbox and access to the administration area inside the requisis_Toolbox. Group must be specified with full DN - as in the example
- `auth.ldap.admin.reserved_active_users: -1`
  Number of user licenses reserved for users in admin.dn (default value: "-1", means no admin license reserved)
- `auth.ldap.supporter.dn: 'CN=ReqTbSupport,OU=ElmAdmins,DC=carcompany,DC=com'`
  Users who are members of this group receive the supporter role for the requisis_Toolbox and access to the administration area inside the ReqIF-Manager tool. Group must be specified with full DN - as in the example.
- `auth.ldap.user_access.dn: 'CN=RmEmployee,OU=ElmDpt,DC=carcompany,DC=com'`
  Users who are members of this group can log in to the requisis_Toolbox, all other users cannot log in. Group must be specified with full DN - as in the example.
- `Auth.ldap.supporter.ldap-filter: 'DCGrp=ReqIFMger.Supporter'`
  Allow activation of Supporter-Role by LDAP-Attribute with Value .
- `auth.ldap.binduser.login.dn 'uid=global,ou=users,dc=carcompany,dc=com'`
  Complete LDAP path (DN) of the technical user. The technical user handles the communication between the requisis_toolbox and the IBM ELM.
- `auth.ldap.binduser.login.pw: 'TechUserPassWord'`
  The password of the technical user.
- `auth.ldap.binduser.searchloginnameby: 'sAMAccountName'`
  Name of the attribute in which the user name is stored. This attribute is used to find the user based on their user name in LDAP (binduser and normal users).

- `auth.ladp.disabled.users.parameter: 'userAccountControl'`
  Name of the attribute in which it is stored whether a user account is inactive. Comment out to deactivate. Only works together with *auth.ldap.disabled.users.parameter.mask*.
- `auth.ldap.disabled.users.parameter.mask: '2'`
  Bit mask against which the attribute from *auth.ladp.disabled.users.parameter* is checked to determine whether a user is inactive. Comment out to deactivate. Only works together with *auth.ladp.disabled.users.parameter*.

4.3.2 Login procedure Diagram

This diagram shows all the parameters required for the login in the *parameters.yml* file. It should help you to find your way around the many parameters. The required parameter names can be found in red on the arrows in the diagram and in the two lists below the diagram. Setting up with an OIDC provider is optional and is shown in blue in the diagram. This can be done later at any time.



Necessary parameters	Additionally required for OIDC
auth.ldap.server	auth.oidc.provider
auth.ldap.port	auth.oidc.client.id
auth.ldap.dc	auth.oidc.client.secret
auth.ldap.user_access.dn	auth.oidc.login.response_type
auth.ldap.supporter.dn	auth.oidc.login.response.mail_attribute
auth.ldap.admin.dn	auth.oidc.login.response.user_attribute
auth.ldap.binduser.login.dn	auth.oidc.login.callbackurl
auth.ldap.binduser.login.pw	

4.3.3 OIDC Settings

OIDC is an authentication layer based on the OAuth 2.0 authorization framework. You can find more information on this at: https://openid.net/ The parameters listed here integrate a second authentication layer into the login. Further information on this can also be found in the chapter 4.3.2 Login procedure Diagram.
If you would like to handle the login process completely via OIDC, please also read subchapter OIDC login. LDAP is not required in this case. After the setup, an OIDC button appears when logging into the *requisis_Toolbox*.

- `auth.oidc.enable: true`
  Switches Authentication via OIDC on/off.
- `auth.oidc.disable_classic_login: false`
  If true: switches off the classic login page and goes directly to the OIDC Auth page. Only switch over after successfully setting up the OIDC login.
- `auth.oidc.provider: 'https://login.microsoftonline.com/<tenentId>/v2.0'`
  The well known URL of the OpenID Server.
  Without the string `'/well-known/openid-configuration'`.
- `auth.oidc.client.id: abc123`
  The OIDC client ID. Generated by OIDC-Provider.
- `auth.oidc.client.secret: 'tbd'`
  The OIDC client secret. Generated by OIDC-Provider.
- `auth.oidc.allow_implicit_flow: true`
  Optional! To use the implicit OIDC flow.
- `auth.oidc.provider_config_params:`
  `- issuer: 'https://login.microsoftonline.com/<tenentId>/v2.0''`
  Optional! Additional provider config parameter as array. Syntax as YAML Spec.
- `auth.oidc.login.url:`
  `'https://login.microsoftonline.com/<tenentId>/oauth2/v2.0/authorize'/`
  Optional! The login URLs endpoint - overwrites the value from the well known URL.
- `auth.oidc.login.response_type: id_token%20token`
  Response type for the login action (valid values: 'id_token%20token', 'code') Check OIDC-Provider settings.
- `auth.oidc.login.response.mail_attribute: 'email'`
  Attribute in which the user's email is included in the OIDC response data, used for ldap search.
- `Auth.oidc.login.response.user_attribute: 'sub'`
  Attribute in which the user's username is included in the OIDC response data - fallback if email not found in ldap.
- `auth.oidc.login.additional_parameter:`
  `'response_mode=form_post&scope=openid+profile+email'`
  Additional parameters which are needed for the login URL. Should not be changed.
- `auth.oidc.login.callbackurl: '<http://localhost:8080>/auth/oidccallback'`
  The callback URL that is called after login.

OIDC login

The following parameters enable authentication by OIDC. If the `auth.oidc.login.response.groups_attribute` is set, the LDAP authentication in chapter 4.3.1 LDAP Login is ignored. If you use the complete OIDC Login, it is currently not possible to use the User Cleanup Script from chapter 7.2.

- `auth.oidc.login.response.mail_attribute: 'email'`
  The attribute containing the user's email within the OIDC response data, which is used for user search.
  `auth.oidc.login.response.user_attribute: 'sub'`
  The attribute containing the user's username in the OIDC response data, used as a fallback if the mail_attribute is not found.
- `auth.oidc.login.response.groups_attribute: ''`
  The attribute in which the user's groups are included in the OIDC response data - if set, OIDC exclusive login is enabled and no LDAP search will be done.
- `auth.oidc.login.response.user_access.group: ''`
  The group which has access to the toolbox - optional
- `auth.oidc.login.response.supporter.group: ''`
  The group to grand user supporter role - optional
- `auth.oidc.login.response.admin.group: ''`
  The group to grand user admin role - optional

4.3.4 Mailer Settings

The parameters listed here must be configured if you want to use the email option of the diskspace warnings feature. Further information on this can be found in the chapter 4.3.11 Configure Diskspace Warnings.

- `mailer_transport: smtp`
  Protocol to be used by the local mail server.
- `mailer_host: 127.0.0.1`
  Host address of your mail server.
- `mailer_port: 9999`
  Port of the mail server.
- `mailer_user: null`
  Username to login on your mail server.
- `mailer_password: null`
  Password of the specified user to login on your mail server.

4.3.5 Misc Settings

- `support.mail: support@company.com'`
  Support email address to be shown at dashboard.
- `Support.text: ''`
  If filled, the default help text in the dashboard will be replaced.
- `testsystem.hosts: ['localhost', '127.0.0.1']`
  Hostnames where the Message “Testsystem” shall be displayed.
- `job.search.maximum.results: 25`
  Default value of results for job search, allowed are: 25, 50, 75, 100, 125, 150
- `jobs.maximum.modules: 10`
  Only <n> modules are processed per job run. If there are more modules in the job than the value entered here, the job is started several times. Lower value = jobs are slower, higher value = more load on the server.
- `parallel.jobs: 4`
  The maximum number of concurrent jobs.
- `parallel.jobs.maximum.modules: 10`
  Run parallel jobs only for jobs with less modules than this value. Used to avoid the server to be blocked by one user exporting/importing huge amount of data for multiple partners at the same time.
- `public.api.key: 123456`
  Shared key for the public API.
- `dng.reqif.check_log_for_error: true`
  Check DNG log for errors.
- `job.notification.uri: 'https://requisis.com/generic_webhook.php'`
  Specify a push URL to send a JSON to as a POST to receive job status changes.
  Parameters: userID, jobId, jobtype, jobStatus, reqifSetId, partnerId
- `notifications.remove.threshold_days: 7`
  Default value is 7, indicating the number of days after which notifications are automatically deleted from the Notification Tab. To disable automatic deletion, set the value to -1.
- `xhtml.container.artifact.type: ''`
  DNG rdf artifact type uri for xhtml container, if empty feature is inactive.

- `xhtml.container.link.type: ''`
  DNG rdf link type uri used for linking xhtml container with artifact, if empty feature is inactive.
- `xhtml.container.link.direction: artifact2xhtmlContainer`
  direction for linking xhtml container with artifact.
  Possible values are: xhtmlContainer2artifact and artifact2xhtmlContainer, artifact2xhtmlContainer is default.
- `close_exportsession.create_baseline: true`
  Show create baseline dialog on close export session. Default setting is true.
- `close_exportsession.remember_baseline: true`
  Show remember baseline dialog for on export session. Default setting is true. Exist only if create baseline dialog is true.
- `close_importsession.create_baseline: true`
  Displays a dialog for creating a baseline when closing an import session.
- `close_importsession.create_baseline_initial_value: true`
  Controls the default behavior of the Create Baseline check box in the Close Import Session dialog box.
  true= Checkbox enabled
- `ui.start_page: dashboard`
  Gives the option of calling up a bundle directly after logging in. Possible values: `dashboard, compare_module, reqif_manager, reqif_viewer`
- `ui.timezone: ''`
  Configure the time zone for all timestamps in the *Requisis Toolbox*; if not configured, the client time zone is used. To configure a specific time zone, use the IANA time zone database name.

4.3.6 ReqIF Manager Settings

- `rm.administration.usermanagement.visible: true`
  Specifies whether the User Management tab should be displayed in the administration area.
- `rm.components_sync.threshold: 1`
  Determines the time span from when a ReqIF Set is considered asynchronous in days.
- `rm.create_log_summary.batch_size: 100`
  Specifies how many job logs of jobs performed by users should be transferred to the summary format within the mysql container. For large amounts of data, the value should be as small as possible to avoid overloading the available memory. -1 deactivates the command.
- `rm.cross_component_links.allow: false`
  Enables the export and import of cross-component links as a Customer between modules in DNG.
- `rm.customer.export.force_view: true`
  In the ReqIF Set role Customer you have to select an explicit view to be able to export a module. Default setting is true.
- `rm.allow.importwithoutmapping: true`
  Allows to select the option *Import without Mapping* when mapping in the role of the Suppliers.
- `rm.disable: false`
  Set to true to hide the ReqIF-Manager in the frontend dashboard.
- `rm.export.changeset.workitem.mandatory_level: 'error'`
  This parameter controls the behavior for exports from Change Sets with mandatory work item and can have the following two values:
  `error` (default): Export aborts with an error.
  `warning`: Export displays a warning. Change Set must be delivered by user.
- `rm.reqifset.owner.domain.blacklist:`
  `- blockedDomain.de`
  `- anotherDomain.com`
  Users with email addresses of the specified domain cannot become owner of a ReqIF Set.
- `rm.force_oslc_import_in_customer_role: false`
  Enforces that imports in the Customer Role can only be made with OSLC.
- `rm.import_mapping.allow_workflow_state_attributes: true`
  Allows mapping of DNG workflow state attributes in import mapping.
- `rm.import_mapping.allow_dng_attribute_creation: true`
  Allows the user to create additional attributes in the mapping dialog.
- `rm.import_mapping.always_use_new_mapping_label: false`
  If true, the new mapping label is always used during import; if false, there is a selection option in partner edit where you can choose between old and new.
- `rm.import_mapping.save_enum_mapping_to_attribute: false`
  Allows you to select a string attribute in the enumeration mapping to save the mapping information.
- `rm.job_cleanup.delete_job_files_after_days: -1`
  Deleting job files after <Value> days, value 0 and lower => no clean up
- `rm.job_cleanup.delete_jobs_after_days: -1`
  Deleting jobs after <Value> days, value 0 and lower => no clean up

- `rm.job_cleanup.keep_last_successful_jobs: -1`
  Keep last <Value> jobs (succesful or with warning) for each partner and jobtype (export/import) also if days are exceeded, value 0 and lower => no exceptions on clean up
- `rm.job.inactivity_timeout_minutes: 60`
  Maximum allowed dastabase inactivity (in minutes) before a job is automatically terminated (e.g., observer jobs without DB updates)
- `rm.notifications.session_close.ui.threshold_days: -1`
  Threshold value for the notification in the frontend (notifications and ReqIF Set List) that a session should be closed (-1 deactivates the functionality). The `app_url` parameter (chapter 4.3) must also be configured
- `rm.notifications.session_close.email.threshold_days: -1`
  Threshold value for the automatic e-mail notification that a session should be closed (-1 deactivates the functionality). The `app_url` parameter (chapter 4.3) must also be configured.
- `rm.notifications.session_close.email.sender: "sender@myCompany.com"`
  Sender of the automatic e-mail notification that a session should be closed. The `app_url` parameter (chapter 4.3 must also be configured.
- `rm.supplier.export.allowModuleAddition: true`
  Boolean flag to control if modules can be added to the export in the supplier role. Set to true to allow module addition, or `false` to restrict it.
- `rm.supplier.import.add_tag_to_deleted_artifact: 'reqif_manager_deleted'`
  During import in supplier role, add the defined tag to artifacts in DNG module that are not included in the imported file, if empty - no tag will be added.

4.3.7 Compare Module Settings

> `cm.job_cleanup.delete_after_days: 14`
This configuration option defines the number of days after which old jobs and debug log files of the compare module are discarded. If the value is set to less than 0, the clean-up command will not be executed.
The clean-up command is executed via crond.php at 21:00/9pm.

> `cm.disable: false`
Show Compare-Module, if true, Compare-Module will not be shown in the dashboard

4.3.8 ReqIF-Viewer Setting

> `rv.disable: false`
Show ReqIF Viewer, if true, ReqIF Viewer will not be shown in the dashboard.

4.3.9 Toolbox Administration Setting

> `admin.disable: false`
Set to true to disable the Toolbox Administration in the frontend dashboard.

4.3.10 Add more detailed user help to dashboard

> support.custom_help_page: true
Switches on clearly visible user help on the dashboard of the requisis_Toolbox.

> support.custom_help_page.button.text: 'Open user help'
Text on the button that calls up the user help page.

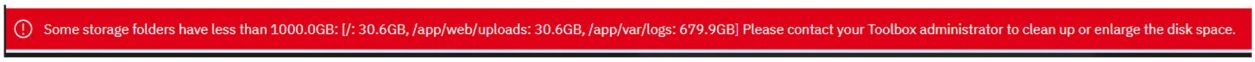
> support.custom_help_page.config:
bundle: reqif_manager
The name of the bundle for which the help is (i.e., reqif_manager compare_module, reqif_viewer)
text: 'For technical problems, please open a ticket, more information at our Supplier Database'
A brief explanation of the linked content.
ankers: [{ text: ReqIF-Manager,
subtext: 'Click here to open a support ticket',
(optional) Text for more precise differentiation from other sources of help.
icon: Email,
(optional) Icon for the anchor (e.g. Email, GroupPresentation, InformationFilled)
link: 'https://my-reqif-manager-kb.com' },
The http(s) address for linking the content.
{ <NextAnker>}]
The anchor(s) for linking to helping content, see parameter descriptions.

4.3.11 Configure Diskspace Warnings

- diskspace.warn.threshold: 10
  Threshold for disk space warning (in GB) - if less space is available warning should be issued.
- diskspace.warn.method:
  - gui
  - email

  Method of disk space warning:
  gui: show warning message in gui (see screenshot below)
  email: send warning email
  null: switched off
- diskspace.warn.email.recipients:
  - admin@mycompany.com

  Recipients for the warning email (null for no recipients), only needed for email method
- diskspace.warn.email.sender: 'diskspacewarnservice@yourdomain.com'
  Sender email address for the warning email, only needed for email method



Example message for the gui method, this is visible to all users of the *requisis_toolbox* and appears at the bottom of the screen. You can also see how much memory the affected folders consume.

4.3.12 No longer used parameters

In this chapter you will find old parameters that are no longer used. If you are still using one of the parameters listed here in your *parameters.yml* file, this can lead to problems. Remove or comment out the corresponding parameters with a preceding #. These parameters are sorted here in alphabetical order - this order does not have to match the actual order in your *parameters.yml* file.

- auth.dngserver.enable
  ~~Is authentication via dng server enabled Default value false~~.
- auth.dngserver.jtsuri
  ~~Address of your jazz team server.~~
- auth.dngserver.prefix
  ~~DNG server prefix. Default value 'dng/~~
- auth.ldap.binduser.enable: true
  ~~Activates the login via a technical user in LDAP~~.
- auth.ldap.company: 'CarCompany'
  ~~Default company name for user if no company name is set in LDAP~~.
- auth.ldap.domain: carcompany
  ~~The Domain of your LDAP server. Will be added to username to authenticate~~.
- auth.ldap.enable: 'true'
  ~~Switch the LDAP authentication~~.
- auth.registration.allowed: true
  ~~To enable this feature set the value to *true* otherwise set to *false.*~~

4.4 Data backup to S3 bucket (optional)

It is possible to set up a backup in an S3 object storage service. To do this, the following parameters must be added to the *.env* file.

> `TOOLBOX_BACKUP_S3_BUCKET='BucketName'`
> # S3 Bucket Name
> `TOOLBOX_BACKUP_S3_ENDPOINT='S3Endpoint'`
> # Endpoint, if not Amazon S3
> `TOOLBOX_BACKUP_S3_REGION='S3Region'`
> # S3 Region
> `TOOLBOX_BACKUP_S3_ACCESSKEY='AccessKey'`
> # S3 Access Key
> `TOOLBOX_BACKUP_S3_SECRET='Secret'`
> # S3 Secret
> `TOOLBOX_BACKUP_S3_FOLDER='Folder'`
> # Folder inside Bucket to store backups

4.5 Customizing the context root

If you would like to run *requisis_Toolbox* behind a reverse proxy (e.g. under the same address as your ELM environment) you will need to setup a context root. The context root is a path that is added directly after the hostname and domain. That will allow you to run *requisis_Toolbox* e.g. under https://elm.company.com/toolbox/

To do this, open the *.env* file in the root directory of your *ReqIF-Manager / requisis_Toolbox* installation and add the parameter: `REQIF_MANAGER_CONTEXTROOT=/<yourContextRoute>`

```
3   MYSQL_ROOT_PASSWORD=[redacted]
4   TZ=Europe/Berlin
5   PHP_DATE_TIMEZONE=Europe/Berlin
6   REQIF_MANAGER_CONTEXTROOT=/toolbox
```

Then save the file and restart the Docker container.

The context root must be added after the hostname and domain. The toolbox can then only be accessed via the complete URI including the specified context root.

4.6 Accessibility of the OIDC through a proxy (optional)

If your infrastructure can only reach your OIDC provider via a proxy server, please also follow the next step. This is not necessary if you do not want to use OIDC or if you can reach your OIDC provider without a proxy server. Add the parameter *http_proxy* to your *.env* file. Use the following syntax for the parameter:

`http_proxy=http://<my-proxy.com>:<port>`

You will find the *.env* file in your main directory of the ReqIF-Manager installation package.

4.7 Change maximum size for file upload

The maximum size for files that can be imported with the ReqIF-Manager is set to 150 MB. To change this size, open the file *.env* from the main directory of the requisis_Toolbox installation package. You must adjust the two parameters *PHP_POST_MAX_SIZE* and *PHP_UPLOAD_MAX_FILESIZE* in this file. If you want to set the maximum size to 500 MB you have to adjust the two parameters as follows:

Example to set max size to 500 MB:

```
1  REQIF_MANAGER_IMAGE=
2  LDAP_IMAGE=
3  MYSQL_ROOT_PASSWORD=
4  TZ=Europe/Berlin
5  PHP_DATE_TIMEZONE=Europe/Berlin
6  PHP_POST_MAX_SIZE=500M
7  PHP_UPLOAD_MAX_FILESIZE=500M
8  #REQIF_MANAGER_CONTEXTROOT=
```

If these two parameters are missing in your *.env* file, you can simply add them yourself. Please note that if you use a reverse proxy server or a load balancer in your infrastructure you must configure them accordingly.

Adjusting the maximum file size above 500MB is absolutely not recommended. The DNG server will be slowed down and react with timeouts, and it will be difficult to work with it.

4.8 Configuration of user-defined DNS entries (optional)

This chapter is intended for users with experience in network configuration.

In containerized applications such as the *requisis_Toolbox*, the etc/hosts file inside the container cannot be used directly, as any changes made there are lost upon container restart. To still enable full integration of the *requisis_Toolbox* into your network environment, you can define custom DNS mappings using the *extra_hosts* section in the *docker-compose.yml* file.

This file is located at the root level of the initial installation package.

To configure *extra_hosts*, you must first determine the required IP addresses.

```
links:
  - 'mysql:mysql'
  - 'ldap:ldap'
extra_hosts:
  - "my.dns.server:192.168.0.111"
volumes:
  - './store/toolbox/config:/store'
  - './store/toolbox/uploads:/app/web/uploads'
  - './store/toolbox/logs:/app/var/logs'
```

The above example ensures that the hostname `my.dns.server` is always resolved inside the container to the IP address `192.168.0.111`.

Note: As with the other parameters in this file, make sure that the indentation is correct.

4.9 Display your company logo above the header (optional)

If you want to display a company logo above the header, you will need the logo to use as a png file. The file must not be higher than 50px (pixels) and the file must have the name *logo.png*. Copy your file to the *store* folder in the Docker container *requisis toolbox*.

The included logo appears in the middle of the title bar of the *ReqIF-Manager* after restarting the Docker Container.

4.10 Changing the background images (optional)

You can customize the background images for the login and logout screens.

Please note the following requirements:

- Minimum resolution: 1024×768 pixels
- Recommended resolution: 1920×1080 pixels (Full HD) or higher
- File format: PNG only

To use your own images:

Copy the image files into the *store* directory and name the files exactly as follows:

For the login screen: *login_background_main.png*

For the logout screen: *logout_background_main.png*

You may also choose to replace only one of the two images. The Container must then be restarted.

4.11 Update of the DNG widget catalog (optional)

Requisis provides a widget to easily start the *ReqIF-Manager / requisis_Toolbox*. To do this, the DNG widget catalog must be updated so that all users can use this widget and subsequent widgets for themselves.

To update the widget catalog, proceed as follows:

- Open any RM-Project page in DNG and click on the gear wheel in the upper right corner and select the option *Manage Server Administration*.
- On the left-hand side of the screen, select the *Advanced Properties* menu item. Navigate to the option *com.ibm.rdm.widgetcatalog.internal.WidgetCatalog* in the RM Client Component area.
- Follow the URL stored there in a new browser tab and copy the complete content into a new file in an editor of your choice. If no URL is stored there yet, define a suitable URL of your choice, and then store your *WidgetCatalog.xml* file there accordingly.
- Open the *requisis_Toolbox* in another browser tab and start the *ReqIF-Manager*. Click on the icon in the top right-hand corner and select the *Widget(s)* option. Click on the button to copy the displayed code for the widget and add it to the file opened in the editor. Make sure that the structure within the file is not destroyed and that the file ends with *</rdf:RDF>* in the last line.
- Save the file in XML format and copy it to the location you specified in the *com.ibm.rdm.widgetcatalog.internal.WidgetCatalog* option. The changes will take effect after approx. 10 minutes.

5 Graphical administration interface

The *requisis_Toolbox* also offers a graphical administration interface. To be able to access this, a corresponding user must be in the LDAP user group that has been stored in the `auth.ldap.admin.dn` parameter in the *parameters.yml* file. Users in this group always have access to the administration interface even if no valid license for a Toolbox product is currently active. To access the administration area after logging in, click on the *Get Started* button in the Toolbox Administration tile or use the app switcher in the top right corner.

The administration area is divided into the following sub-areas:

- *Users*: Overview of users with supporter and admin rights
- *Errors*: Errors that have occurred in the tools
- *Maintenance*: Maintenance mode on/off
- *Parameters*: Display of the current parameter settings in the *parameters.yml* file
- *Licenses by user*: Overview of the licensed tools and their license consumption.
- *LogFile Viewer*: A tool for analyzing the generated log files

You will find a corresponding chapter for each sub-area here.

5.1 Users

Here you can see the two lists of users with supporter rights and users with admin rights. The supporters can be controlled via the parameter `auth.ldap.supporter.dn` and the admins via the parameter `auth.ldap.admin.dn`. The difference between these two user groups is that Supporter users only see the administration area in the *ReqIF-Manager* tool. Admin users also see the *Toolbox Administration* area in the dashboard of the *requisis_Toolbox*.

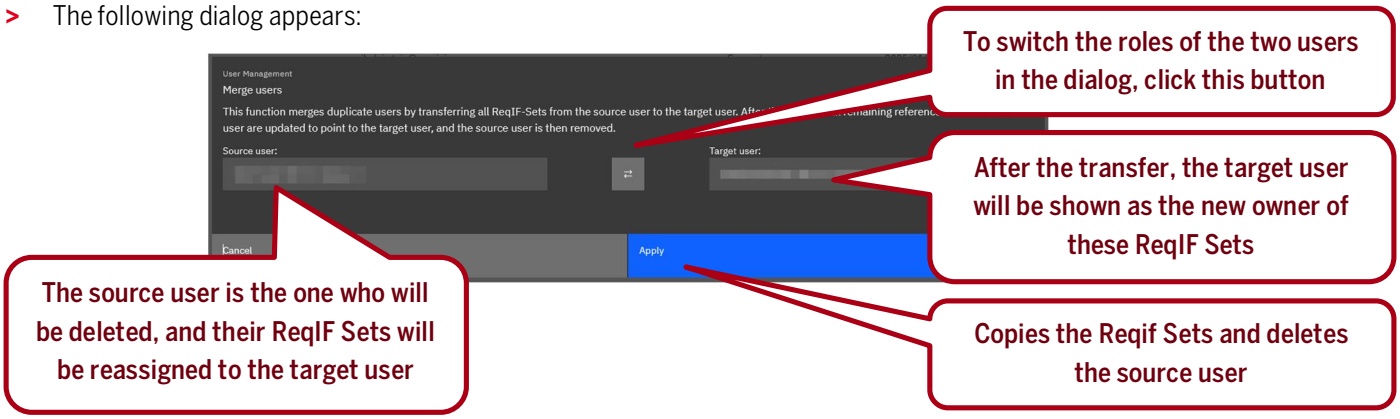
5.1.1 Merge duplicate Users

With this function it is possible to copy ReqIF Sets from a duplicate user to an original user and then delete the duplicate user. The original user becomes the new owner of the duplicate user's ReqIF Sets and the original user also becomes the creator of the duplicate user's jobs.

Important Note:
Before using this function, make sure that you are dealing with a (one) physical person with two accounts.
This function is not intended to remove inactive users. Read chapter 8.2 in the *ReqIF-Manager* user manual on this topic.

To do this, proceed as follows:

> Use the checkboxes in the list to select two corresponding users, regardless of the order.
> Click on the *Merge users* button in the top right-hand corner
> The following dialog appears:



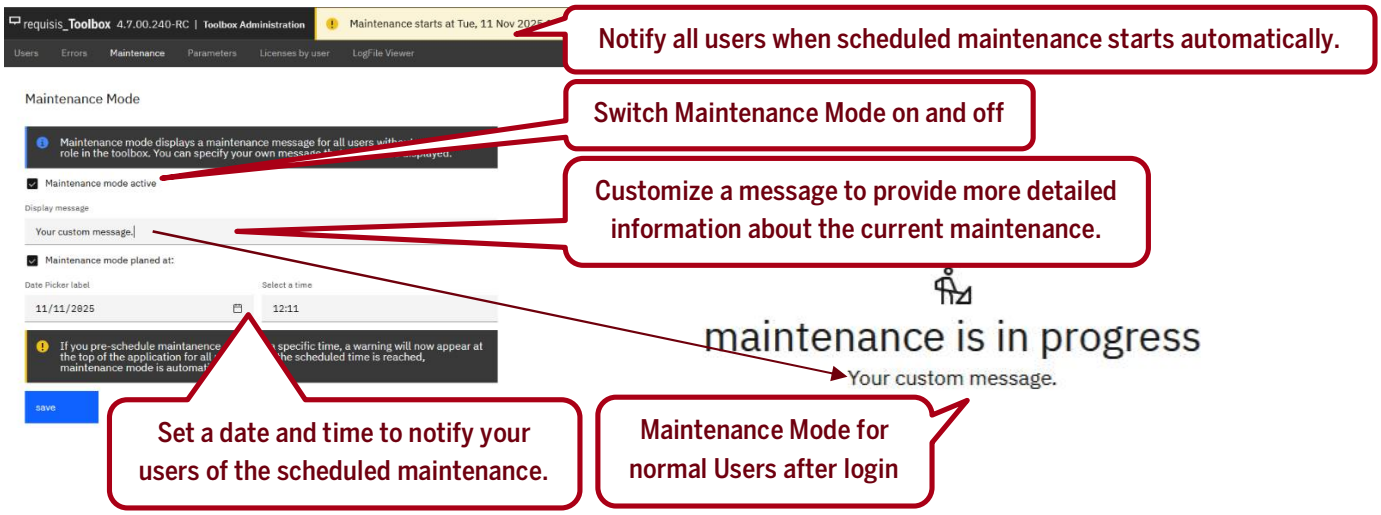
> Click on the Apply button to copy the ReqIF Sets as configured and then delete the user that was specified in the Source user field.

5.2 Errors

The *Errors* tab shows you a list of the last errors that occurred when using your licensed tools from the *requisis_Toolbox*. For a complete error message click on ⌄. When contacting support regarding an error, please provide support with this information. In the top right-hand corner, you will also find a button for copying the error message.

5.3 Maintenance

In the *Maintenance* tab, you have the option of switching the maintenance mode on and off. Normal users cannot use the *requisis_Toolbox* until this is done. In the *Display message* field, you can formulate a more detailed message to inform your users more precisely. Optionally, you can also set a *date* and *time* for the scheduled maintenance to start, giving your users plenty of warning. The maintenance will then start automatically at the set time. Then click on the *save* button to display this message below the maintenance mode notice for your normal users after logging in.



Jobs that are already running are still processed after maintenance mode is switched on. Administrators still have unrestricted access when maintenance mode is switched on.

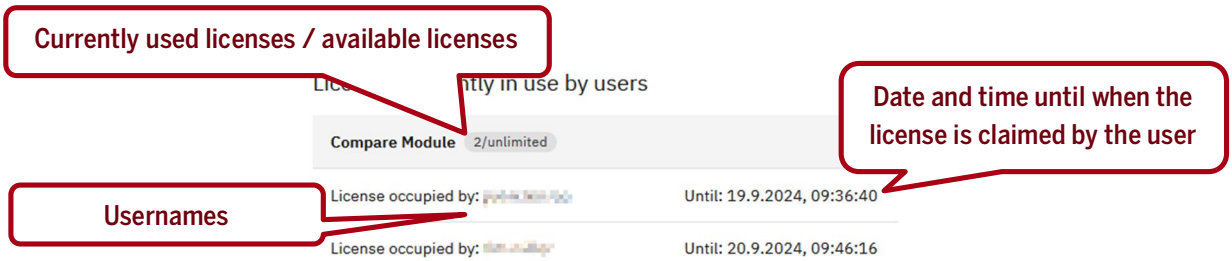
5.4 Parameters

This tab shows the current configuration of the `parameters.yaml` file from the subfolder store/toolbox/config. This should help you to conveniently view and check the current parameters. However, **changes cannot be made here - these must be made directly in the file** as usual. An explanation of each parameter can be found in chapter 4.3 Settings in parameters.yml of this manual.

If you want to download the currently used *parameters.yml* file, click on the ⬇ icon in the upper right corner.

5.5 Licenses by user

This tab shows the license usage for each licensed tool in the *requisis_Toolbox*. For each tool, you can see a list of users who are currently claiming a license. On the right-hand side of the screen, you can also see when the license is no longer being used by the user. Each login of a user blocks a license for 7 days from login.



5.6 LogFile Viewer

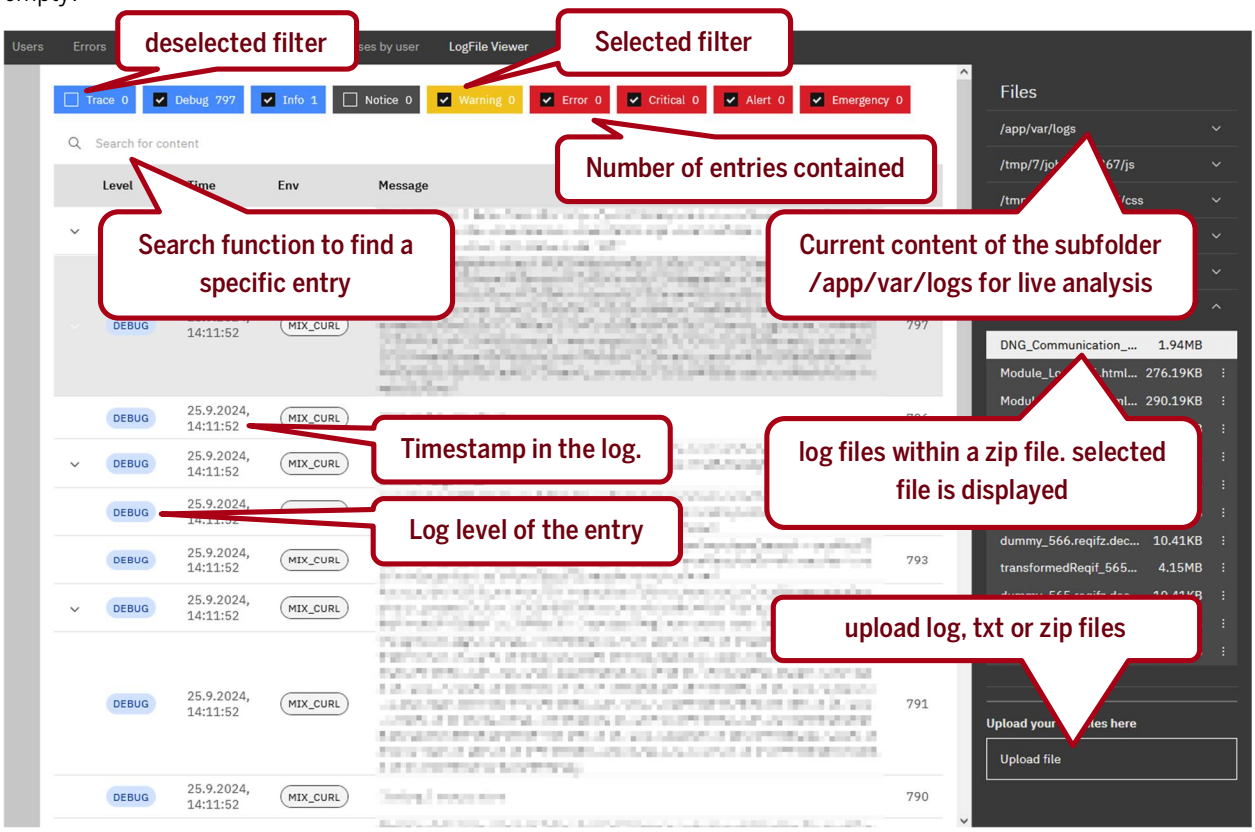
In the LogFile Viewer tab, you can upload, view and analyze files with the extension *.log, .txt* or *.zip*. If you upload a zip file, only files with the extension .log are displayed in the list of contained files.

Important Note:
If you try to view an incompatible file, you will receive the following output in the *Message* column:
Logfile is not compatible
These are primarily files that are not in monolog format.

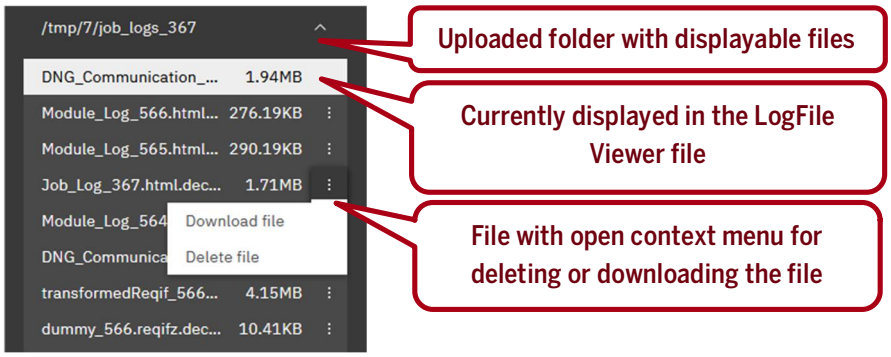
In the top right-hand corner, you will find the *Files* area. Here you can view the contents of the `/app/var/logs` subfolder, which contains all the log files for the *requisis_toolbox*. Live analyses of the requisis_toolbox_login.log, for example, are therefore possible.

You also have the option of uploading and analyzing your own files here using the *Upload file* button, for example debug logs from the *ReqIF-Manager*. To display the file, click on the desired entry in the *Files* area and click again to display the corresponding file. The selected file is displayed as a list of log entries. If you are looking for a specific entry or string, you will find a search function above the table headings. The most recent entry is displayed first in the list to find errors quickly.

It is possible to filter by log level. You can see how many log entries there are in the currently selected file by the number of the respective filter. Click on the corresponding filter to select or deselect it. To customize the list of existing entries according to your wishes. If a file is selected for viewing that does not contain any of the active filters, the table remains empty.



To remove files from the list, click on the ⋮ menu icon to the right of the corresponding file. If you want to remove a complete folder from the *LogFile Viewer*, right-click this folder and select the *Delete Folder* option. This option is not available for the `app/var/logs` folder.



If you delete a file, it is deleted in the *LogFile viewer*. The corresponding file remains on its data source (e.g. local disk drive).

6 Build Docker Containers

There are several ways to build Docker containers. Choose the variant that fits best for your infrastructure. For evaluation purposes we recommend the included command line scripts.

6.1 Start Docker using command line script

To start the Docker Container, we have added a corresponding script to the initial zip package for Windows and Linux systems. Once you run the startDocker scripts the ReqIF Manger will be installed. This will take several minutes.

- Run the *startDocker.cmd* script at Windows Systems.
- Run the *startDocker.sh* script at a Linux System.

6.2 Start Docker using docker-compose

The default `docker-compose.yml` defines how the docker-containers shall be started on your machine / server.

It can be customized to your needs (e.g. changing port numbers or binding the docker to a different network interface)

Among other things, the storage locations for the generated log files, uploaded reqifz files etc. can be customized. To do this, navigate to the *services* -> *server* -> *volumes* area of the file.

Default storage location paths:

- `./store/toolbox/config`
- `./store/toolbox/uploads`
- `./store/toolbox/logs`

When the default settings in `docker-compose.yml` are meeting your requirements, you can start the docker containers by navigating into the folder where the `docker-compose.yml` is stored and type:

```
docker-compose up
```

if you want to run the container in foreground

or

```
docker-compose up -d
```

if you want the docker container to be run in background.

To stop the container, you can use

```
docker-compose stop
```

if you only want to stop the container

or

```
docker-compose down
```

if you want to stop and destroy the container. All data is placed outside the container in the /store folder, so it is safe to dispose of the container.

6.3 Using own docker deployment frameworks

If you want to deploy the docker container in other deployment frameworks like on AWS you can use the docker-compose.yml as a kind of description how the docker containers are interconnected and which data stores need to be mounted.

6.4 Using Kubernetes for deployment

We are offering a helm-chart for deployment on Kubernetes, please contact us for more details.

7 Cron Commands

The cron commands must be configured within the container. If you are not using a graphical user interface for this, use the command:
`docker exec -it <placeholder_Toolbox_ServerContainerName> /bin/bash`
to access the container's console. You can leave this with the `exit` command.

7.1 Job Cleanup Script

You have the option to use a cleanup script that deletes old jobs to save disk space. To do this, open a console on the *Requisis toolbox server* Docker container. You run the script with the following command:

```
php bin/console requisis:jc:job-cleanup <fileDays> <jobDays>
```

>	`fileDays:`	Deletes all files of all jobs (user independent) that are older than 30 days (default).
>	`jobDays:`	Deletes all jobs of all users of the last 60 days (default).

Depending on the number of files and jobs to be deleted, the execution may take several minutes. As soon as the script is finished it reports with the message: *finished jobcleanup script*.

7.2 User Cleanup Script

It is possible to query the Active Directory via LDAP with a cron job for inactive/locked users. To start this cron job, open a console on the *Requisis toolbox server* Docker container. The command to start the job is:

```
php bin/console requisis:rm:users-cleanup
```

If a user wants to log in to the requisis_Toolbox who was locked via the cron job and is no longer inactive/locked via LDAP, the account is automatically unlocked again at login.

7.3 Transfer job logs to save memory space

It is possible to reduce the amount of memory used by the mysql database container. To do this, the job logs of the users are converted into a summary format. To adjust the number of summarized jobs, you can adjust the parameter `rm.create_log_summary.batch_size` in the *parameters.yml* file. You run the script with the following command:

```
php bin/console requisis:rm:create-log-summary
```

The oldest jobs that have not yet been converted are transferred each time this command is executed. The user does not notice the transfer, all features are retained.

7.4 Support to move DNG domain

After changing the ELM/DNG domain, you must use the following command so that all database entries of the *requisis_Toolbox* are adjusted correctly. To do this, open a console on *the Requisis toolbox server* container. You run the script with the following command:

```
php bin/console requisis:cd:changeDomain <OLD-DAMAIN> <NEW-DOMAIN>
```

>	OLD-DAMAIN:	The old domain that is currently still in use. (ex. old.ELM.com).
>	NEW-DOMAIN:	The new domain to be used in the future. (ex. new.ELM.com)

The execution should normally be completed in several minutes but can increase exponentially depending on the amount of data.

8 Update procedure for new release

Deployment of new versions of ReqIF-Manager is done via our Docker container distribution system Harbor. When you licensed ReqIF-Manager, you received an account that logs into Harbor to update your ReqIF-Manager.

8.1 Create mySQL Backup

See backup chapter in this manual.

8.2 Shutdown requisis-toolbox Docker Container

To shut down the requisis-toolbox container by open a command line / shell and type:

```
docker-compose stop
```

8.3 Check the .env file

You find this file on the top level of the Docker container. In the .env file settings for updating the ReqIF-Manager are managed. As well as the used time zones of the ReqIF-Manager are determined.

You can adjust the following parameters:

- COMPOSE_PROJECT_NAME=requisis_toolbox
  Is used to specify the name of the Docker project, as a prefix for the names of the containers and volumes started by docker-compose.yml
- REQIF_TOOLBOX_IMAGE=registry.requisis.com/requisis/requisis-toolbox:latest-stable
  Here you can determine which version should be pulled from the harbor. You can specify an explicit version such as `2.9.03-stable`. Or if you want automatic updates for your version, use the tag `2.9-stable` to get the latest version of the ReqIF-Manager 2.9.
- LDAP_IMAGE=registry.requisis.com/requisis/requisis-toolbox/ldap-docker:latest-master
  Automatic updating of the LDAP Docker Container.
- MYSQL_ROOT_PASSWORD=start.root
  Adjust the MySQL password.
- TZ=Europe/Berlin
  Adjust the time zone for OS.
- PHP_DATE_TIMEZONE=Europe/Berlin
  Adjust the time zone for PHP.
- HARBOR_USER=<yourHarborRobotUser>
  The username of your Harbor account to update the ReqIF-Manager.
- PW=<veryLongPassword>
  The password for your harbor account to login.

The *.env* file also contains parameters for configuring an S3 bucket. Read chapter 4.4 Data backup to S3 bucket (optional)

8.4 Start new Docker Container

To start the ReqIF-Manager open a command line / shell and run the script *startDocker.cmd* it will check if you are currently using the latest version configured in your *.env* file and update it if not. Afterwards, the Docker containers are started up again. This can take some time due to the update. This depends on how many DNG modules are used. As soon as you can see the interface of the ReqIF-Manager in the browser again, the update is finished.

With the command `docker-compose logs -f server` you can follow the process of update and migration.

8.5 Check migration logs

After the container is running again, check the log files to see if there were any issues during database migration.

Please check that your database was updated correctly by checking the content of:

migraions-schema-update-needed.log in folder store/toolbox/logs/app/

If there were no issues, you will find the following text in the log file:

```
[OK] Nothing to update - your database is already in sync with the current entity metadata.
```

In case that differences are shown here. Please contact support and provide the content of

migrations.log in folder store/toolbox/logs/app and the database backup that has been created before.

Do not use the ReqIF-Manager before

9 API

The *requisis ReqIF-Manager* provides an API. You can explore the available API functions using a Swagger editor like https://editor.swagger.io/. You can access the API documentation through the *About* dialog, located in the upper-right ⋮ menu of the *ReqiF-Manager* application. The dialog contains a direct link to the API documentation.

Important: To use the API, the following parameters must be configured in the *parameters.yaml* file:

- All `auth.ldap.binduser.*` parameters must be configured and enabled.
- The value of the `public.api.key` parameter must be changed from the default `123456` to a secure key.
- The `app_url` parameter must be set with the base URI (and context root, if applicable) under which the *requisis_Toolbox* is accessible.

9.1 Shared secret calculation

The API uses a time-based hashed API key for authentication. The client must generate this hash and send it with each API request. The shared secret is composed of:

- the public API key
- a *salt* derived from the current timestamp, which changes every 5 minutes

The server accepts hashes generated with the *salt* of the current, previous, and next time period to account for slight clock differences between the client and the server. Here is an example in `JavaScript` of how the shared secret hash is created:

```
// Import the required library
const crypto = require('crypto-js');


// Value of the parameter public.api.key
const publicApiKey = '123456';


// Calculate the salt (current timestamp in 5-minute blocks)
const salt = Math.floor(Date.now() / 1000 / 300);


// Create the API key (SHA-256 hash) and set the environment variable for the
API key
const apiKey = crypto.SHA256(publicApiKey + salt).toString(crypto.enc.Hex);
```

9.2 Impersonation User

The impersonation User is the user in whose name the API requests are executed. If the user is not already existing the user information will be taken from LDAP.
Header information for API request:

```
impersonation-user: <userNameOfUserToPerformActionsAs>
```

10 Backup and Restore Scripts

We recommend backing up the Docker images so that you can fall back on a specific version instead of always using the latest version. Copy the following two scripts into an editor of your choice and save them as .sh files (linux) or .cmd file (Windows). Add the necessary information for:

<PLACEHOLDER_REQIF-MANAGER-CONTAINERNAME>

<PLACEHOLDER_MYSQL-CONTAINERNAME>

<PLACEHOLDER_MSQL-ROOT-PW

To get running docker container names you can use the command: docker ps -a

Give yourself with the command chmod 777 <filename>.sh execution rights for the scripts. Then start the backup script with the command: ./backupReqifMGR.sh (if your filename is "backupReqifMGR.sh") This takes some time depending on the size of the data to be backed up. The backed-up data is in the file backup.tar.gz (in your current folder from where you started the script).

After that you can use the backups for restoring in your new environment.

10.1 Linux backup script

```
#!/bin/bash

export REQIF_MGR_CONTAINER=<PLACEHOLDER_REQIF-MANAGER-CONTAINERNAME>
export MYSQL_CONTAINER=<PLACEHOLDER_MYSQL-CONTAINERNAME>
export MYSQL_ROOT_PASSWORD=<PLACEHOLDER_MSQL-ROOT-PW>

#backup mysql
docker exec -ti $MYSQL_CONTAINER mysqldump -A -C -x --max-allowed-packet=2G
-u root -p$MYSQL_ROOT_PASSWORD --result-file=dump.sql
docker exec -ti $MYSQL_CONTAINER gzip dump.sql
rm dump.sql
docker cp $MYSQL_CONTAINER:dump.sql.gz dump.sql.gz
docker exec -ti $MYSQL_CONTAINER rm dump.sql.gz

#backup reqif files
docker exec -ti $REQIF_MGR_CONTAINER tar -zcvf backup.tar.gz
/app/web/uploads
rm backup.tar.gz
docker cp $REQIF_MGR_CONTAINER:/app/backup.tar.gz backup.tar.gz
docker exec -ti $REQIF_MGR_CONTAINER rm /app/backup.tar.gz
```

10.2 Windows backup script

```
SET REQIF_MGR_CONTAINER=<PLACEHOLDER_REQIF-MANAGER-CONTAINERNAME>
SET MYSQL_CONTAINER=<PLACEHOLDER_MYSQL-CONTAINERNAME>
SET MYSQL_ROOT_PASSWORD=<PLACEHOLDER_MSQL-ROOT-PW>

rem backup mysql
docker exec -ti %MYSQL_CONTAINER% mysqldump -A -C -x --max-allowed-packet=2G
```

```
-u root -p%MYSQL_ROOT_PASSWORD% --result-file=dump.sql
docker exec -ti %MYSQL_CONTAINER% gzip dump.sql
del dump.sql
docker cp %MYSQL_CONTAINER%:dump.sql.gz dump.sql.gz
docker exec -ti %MYSQL_CONTAINER% rm dump.sql.gz

rem backup reqif files
docker exec -ti %REQIF_MGR_CONTAINER tar -zcvf backup.tar.gz
/app/web/uploads
del backup.tar.gz
docker cp %REQIF_MGR_CONTAINER%:/app/backup.tar.gz backup.tar.gz
docker exec -ti %REQIF_MGR_CONTAINER% rm /app/backup.tar.gz
```

10.3 Linux restore script

The restore script must be in the same folder as your backup.tar.gz file with the data you want to restore. To restore the backed-up data use the following bash script:

```
#!/bin/bash

export REQIF_MGR_CONTAINER=<PLACEHOLDER_REQIF-MANAGER-CONTAINERNAME>
export MYSQL_CONTAINER=<PLACEHOLDER_MYSQL-CONTAINERNAME>
export MYSQL_ROOT_PASSWORD=<PLACEHOLDER_MSQL-ROOT-PW>

#restore mysql
docker cp dump.sql.gz $MYSQL_CONTAINER:dump.sql.gz
docker exec -ti $MYSQL_CONTAINER gunzip dump.sql.gz
docker exec -ti $MYSQL_CONTAINER bash -c 'mysql --max-allowed-packet=2G -u
root -p$MYSQL_ROOT_PASSWORD <dump.sql'
docker exec -ti $MYSQL_CONTAINER rm -f dump.sql

#restore reqif files
docker cp backup.tar.gz $REQIF_MGR_CONTAINER:/app/backup.tar.gz
docker exec -ti $REQIF_MGR_CONTAINER tar -xvf /app/backup.tar.gz -C /
docker exec -ti $REQIF_MGR_CONTAINER rm -f /app/backup.tar.gz
```

10.4 Windows restore script

The restore script must be in the same folder as your backup.tar.gz file with the data you want to restore. To restore the backed-up data use the following bash script.

```
export REQIF_MGR_CONTAINER=<PLACEHOLDER_REQIF-MANAGER-CONTAINERNAME>
export MYSQL_CONTAINER=<PLACEHOLDER_MYSQL-CONTAINERNAME>
export MYSQL_ROOT_PASSWORD=<PLACEHOLDER_MSQL-ROOT-PW>

rem restore mysql
docker cp dump.sql.gz %MYSQL_CONTAINER%:dump.sql.gz
docker exec -ti %MYSQL_CONTAINER% gunzip dump.sql.gz
docker exec -ti %MYSQL_CONTAINER% bash -c 'mysql --max-allowed-packet=2G -u
root -p$%MYSQL_ROOT_PASSWORD% <dump.sql'
docker exec -ti %MYSQL_CONTAINER% rm -f dump.sql

rem restore reqif files
docker cp backup.tar.gz %REQIF_MGR_CONTAINER%:/app/backup.tar.gz
docker exec -ti %REQIF_MGR_CONTAINER% tar -xvf /app/backup.tar.gz -C /
docker exec -ti %REQIF_MGR_CONTAINER% rm -f /app/backup.tar.gz
```

11 Glossary

Concept	Meaning
Baseline	A baseline is a frozen snapshot of a stream that can no longer be changed.
Change set	A change set is the smallest change unit in DNG. The ReqIF-Manager automatically creates change sets for carrying out the import. These can either be delivered to the stream or discarded by the user after the import.
(Configuration) Context	In DNG it is possible to create different contexts to perform different tasks or processes. Possibilities for contexts in DNG include Streams, Change Sets, Baselines or Global Configurations in which artifacts, folders, etc. can reside. The ReqIF-Manager receives its data (artifacts, modules, folders) from the corresponding configuration context. For this reason, it is very important to provide the correct context so that the export and import process runs correctly. Currently it is only possible to select the context from a Stream or Change Set
Customer (role)	In the customer role, you export requirements and only import the values of the harmonization attributes of your harmonization process that you have received from the supplier.
Exchange partner	An exchange partner is a company or a department from which you receive exports from a different Requirements Management System, which you then import with the ReqIF-Manager. In return, you export data from DNG using the ReqIF-Manager, which your exchange partner then imports.
Job	When you start an import or export, a job with the included modules is created in the DNG server queue. You can track the progress while the job is being carried out, and then examine the logs; for export jobs, you can then download the modules as a ReqIFz file.
Mapping	During mapping, you map the data model from your exchange partner onto your own. This occurs in ReqIF sets with the supplier role when you import modules for the first time. The mapping is subdivided into the sub-stages type mapping, attribute mapping and link mapping. In these sub-stages, you decide which data

Concept	Meaning
	you would like to import and which types or attributes you would like to use for this.
Module (module artifact)	In DNG, artifacts can be consolidated into module artifacts. For simplicity, this manual uses the word "module" from the classic Doors for the module artifacts from DNG.
ReqIF/Reqifz (file format)	ReqIF stands for Requirements Interchange Format. This is an XML file format that assists in the exchange of all related metadata between software tools such as the ReqIF-Manager and other manufacturers.
ReqIF set	In a ReqIF set, you compile various module quantities from various DNG projects and components. In the ReqIF set, you also define which modules and attributes an exchange partner may receive and edit. They initiate imports to and exports from the DNG database.
(Export/Import) session	Before you can configure an export or import for an exchange partner, you must start a session. During import sessions, all imports are imported into the same change set. At the end of export sessions, you can create a baseline for all the involved DNG components and have the ReqIF-Manager “remember” this baseline for all successfully exported modules
Stream	A stream can be compared with a branch from software development. The artifacts can be edited in a stream. In a stream, you can create baselines that take a snapshot of the stream.
Super set	A super set are modules that are included for all Exchange Partners of a ReqIF Set. Super set modules can be configured centrally for all exchange partners of the ReqIF set
Supplier (role)	As a supplier, you import all the data from the customer and map it onto your data model. You only export the attributes defined in the harmonization process.

requisis_**Toolbox**

requisis *Outstanding in Engineering*

REQUISIS GmbH
Walter-Benjamin-Platz 8, D-10629 Berlin, Germany
Telefon +49 30 / 53 65 06 - 700; Fax +49 30 / 53 65 06 - 300

REQUISIS GmbH
Willy-Brandt-Str. 54, D-70173 Stuttgart, Germany
Telefon +49 711 / 528 529 - 700; Fax +49 711 / 528 529 - 300

REQUISIS GmbH Product Support
DE-Hotline +49 30 536 506 - 666
US-Hotline 1-844-REQUISI, ext. 666

product-support@requisis.com, **https://requisis.com**