



Instructions for the Installation and Configuration of requisis_Toolbox Version 4.5

requisis_Toolbox

Table of Contents

Table of Contents.....	2
1 Introduction	5
2 Requirements	5
3 Installation	5
3.1 Download the requisis_Toolbox with the requisis_ReqlF Manger	5
3.2 Description of Package and Container	5
3.2.1 Required Containers	5
Mysql.....	5
Requisis-toolbox.....	5
3.2.2 Mounts.....	6
mysql	6
toolbox-server	6
3.2.3 File and Folder Structure of deployment package	6
3.3 Setup initial package	7
4 Configuration	8
4.1 Configuration of DOORS Next Generation.....	8
4.2 Configuring DNG-Servers in mix-config.json file.....	9
4.2.1 SSL-client certificates for DNG Server Connections (optional)	11
4.2.2 Warning: There are more DNG servers configured than licensed.	11
4.3 Settings in parameters.yml	12
4.3.1 LDAP Login.....	13
4.3.2 Login procedure Diagram.....	14
4.3.3 OIDC Settings.....	15
4.3.4 Mailer Settings	16
4.3.5 Misc Settings.....	16
4.3.6 ReqlF Manager Settings	18
4.3.7 Compare Module Settings.....	19
4.3.8 ReqlF-Viewer Setting.....	19
4.3.9 Add more detailed user help to dashboard	19
4.3.10 Configure Diskspace Warnings	20
4.3.11 No longer used parameters.....	20
4.4 Data backup to S3 bucket (optional)	21

requisis_Toolbox

4.5	Customizing the context root.....	21
4.6	Accessibility of the OIDC through a proxy (optional).....	21
4.7	Change maximum size for file upload.....	22
4.8	Display your company logo above the header(optional)	22
4.9	Update of the DNG widget catalog (optional)	23
5	Graphical administration interface.....	24
5.1	Users	24
5.2	Errors.....	24
5.3	Maintenance.....	25
5.4	Parameters	25
5.5	Licenses by user	25
5.6	LogFile Viewer	26
6	Build Docker Containers.....	28
6.1	Start Docker using command line script	28
6.2	Start Docker using docker-compose	28
6.3	Using own docker deployment frameworks	29
6.4	Using Kubernetes for deployment	29
7	Cron Commands.....	30
7.1	Job Cleanup Script.....	30
7.2	User Cleanup Script.....	30
7.3	Support to move DNG domain	30
8	Update procedure for new release.....	31
8.1	Create mySQL Backup.....	31
8.2	Shutdown requisis-toolbox Docker Container	31
8.3	Check the .env file.....	31
8.4	Start new Docker Container	31
8.5	Check migration logs	32
9	API	33
9.1	Shared secret calculation.....	33
9.2	Impersonation User	33
10	Backup and Restore Scripts.....	34
10.1	Linux backup script.....	34
10.2	Windows backup script.....	35
10.3	Linux restore script.....	35

requisis_**Toolbox**

10.4	Windows restore script.....	36
11	Glossary	37

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_ReqIF-Manager* the following conditions are required:

- > IBM DOORS Next Generation 6.0.6 or later
- > Local Installation of Docker Engine or Docker on a server and Docker compose
- > Delivery Package of *requisis_ReqIF-Manager* provided by REQUISIS

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 certificates 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*.

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 provide 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/URLs 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

requisis_Toolbox

Path and File	Description
store\toolbox\config\parameters.yml	Contains the configuration parameters of requisis_Toolbox
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 be modified.
store\toolbox\config\routing.yml	Contains URL path parameters. This file will be provided by REQUISIS and must not be modified.
store\toolbox\config\init	This file indicates that the store has been initialized. This file will be provided by REQUISIS and must not be 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
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.

requisis_Toolbox

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

The following page should open:

OAuth Consumers ?

Use this page to manage your OAuth consumers or to generate keys for new consumers. An OAuth consumer is an application, a Jazz Team Server, or a website that is allowed to send requests to this Jazz Team Server. Consumers sign their requests with an authorized consumer key and secret. For applications registered in the Jazz Team Server setup wizard, consumer keys were generated automatically. For requests from other Jazz Team Servers, you must generate a consumer key and secret that then require authorization. These requests that require authorization are listed in the Provisional Key section below. The Authorized Key section lists all authorized consumer keys. You can add, update, or remove consumer keys. Trusted consumers will be able to share their authorization with other trusted consumers. Websites and applications external to your organization should be prompted for approval to access data.

Register Consumer

Use the form below to generate an authorized OAuth consumer key.

Property	Value
Consumer Key	1 The key will be generated by the server Click here to pick the consumer key instead.
Consumer Name	2 requisis_dng_migration
Consumer Secret	3
Re-type Consumer Secret	4
Trusted	5 <input checked="" type="checkbox"/>

Register

- 1) Set **Consumer Key**: Either choose one or note down the key generated by the server.
- 2) Set **Consumer Name**: Set to "requisis_ReqIF Manager" (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**

requisis_Toolbox

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
<i>Consumer Key</i>	oauth_clientId
<i>Secret Key</i>	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.



requisis_Toolbox

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.

requisis_Toolbox

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.

requisis_Toolbox

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
- > `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

requisis_Toolbox

4.3.1 LDAP Login

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

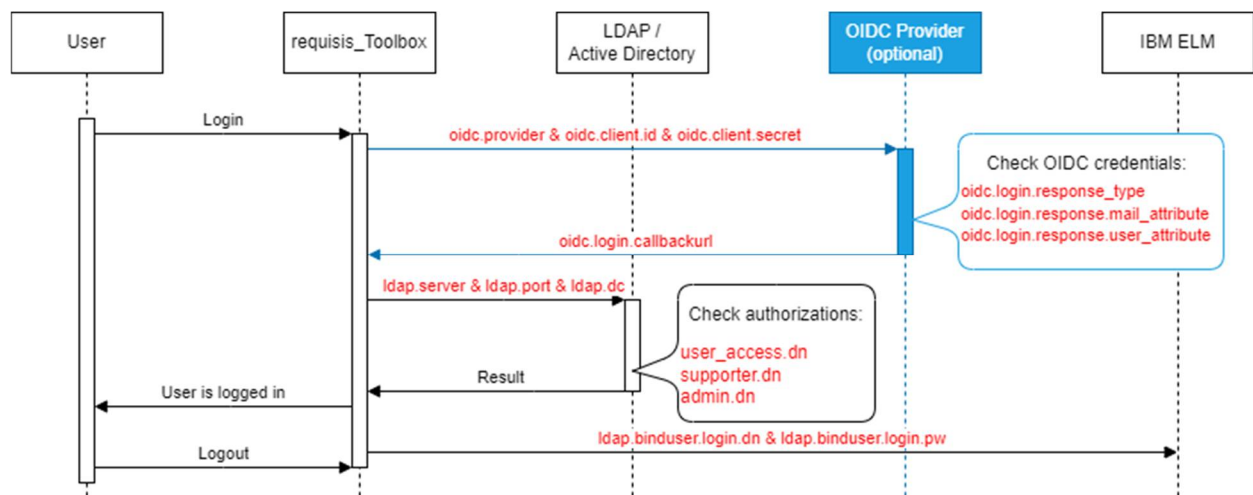
- > `auth.ldap.debug: 'false'`
If the value is set to 'true', the LDAP debug mode and the login.log are activated for more detailed error messages. This helps you to analyze problems with the login. You can also find the requisis_toolbox_login.log in the subfolder store/toolbox/logs
- > `auth.ldap.server: '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.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*.

requisis_Toolbox

- > `auth.ldap.disabled.users.parameter.mask: '2'`
 Bit mask against which the attribute from `auth.ldap.disabled.users.parameter` is checked to determine whether a user is inactive. Comment out to deactivate. Only works together with `auth.ldap.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
<code>auth.ldap.server</code>	<code>auth.oidc.provider</code>
<code>auth.ldap.port</code>	<code>auth.oidc.client.id</code>
<code>auth.ldap.dc</code>	<code>auth.oidc.client.secret</code>
<code>auth.ldap.user_access.dn</code>	<code>auth.oidc.login.response_type</code>
<code>auth.ldap.supporter.dn</code>	<code>auth.oidc.login.response.mail_attribute</code>
<code>auth.ldap.admin.dn</code>	<code>auth.oidc.login.response.user_attribute</code>
<code>auth.ldap.binduser.login.dn</code>	<code>auth.oidc.login.callbackurl</code>
<code>auth.ldap.binduser.login.pw</code>	

requisis_Toolbox

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/> If you want to use OIDC, you must not change the LDAP settings as certain properties of the user are still required. The user directory of OIDC is not used. OIDC is only used for authentication. LDAP is still required to authorize users.

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/<tenantId>/v2.0'`
The well known URL of the OpenID Server.
- > `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/<tenantId>/v2.0'`
Optional! Additional provider config parameter as array. Syntax as YAML Spec.
- > `auth.oidc.login.url:`
`'https://login.microsoftonline.com/<tenantId>/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.

requisis_Toolbox

4.3.4 Mailer Settings

The parameters listed here must be configured if you want to use the email option of the disk space warnings feature. Further information on this can be found in the chapter 4.3.10 Configure Disk Space 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 more, the job will be started several times.
- > `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 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
- > `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.

requisis_Toolbox

- > `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`

requisis_Toolbox

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.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.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.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.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.

requisis_Toolbox

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 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.

`anchors: [{ 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.

requisis_Toolbox

4.3.10 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

ⓘ Some storage folders have less than 1000.0GB: [/: 30.6GB, /app/web/uploads: 30.6GB, /app/var/logs: 679.9GB] Please contact your Toolbox administrator to clean up or enlarge the disk space.

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.

4.3.11 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**.~~



requisis_Toolbox

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=mysql_root_password
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.

requisis_Toolbox

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=/var/lib/docker/volumes/reqif_manager_image/_data/reqif-manager-image.png
2  LDAP_IMAGE=/var/lib/docker/volumes/ldap_image/_data/ldap-image.png
3  MYSQL_ROOT_PASSWORD=root
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=/var/lib/docker/volumes/reqif_manager_contextroot/_data/reqif-manager-contextroot.png
```

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 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.

requisis_Toolbox

4.9 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.

requisis_Toolbox

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*.

5.2 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.

requisis_Toolbox

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. Then click on the **save** button to display this message below the maintenance mode notice for your normal users after login.

The screenshot shows the 'Maintenance Mode' configuration page. It includes a 'Maintenance Mode' section with a checkbox for 'Maintenance mode active' and a 'Display message' text area. A 'save' button is at the bottom left. A large message 'maintenance is in progress' is shown on the right, with a 'Your custom message.' label below it. Callouts point to the 'Maintenance mode active' checkbox, the 'Display message' field, and the 'maintenance is in progress' message.

Switch Maintenance Mode on and off


Customize a message to provide more detailed information about the current maintenance.

Maintenance Mode for normal Users after login

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.yaml` of this manual.

If you want to download the currently used `parameters.yaml` 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.

The screenshot shows the 'Licenses by user' interface. It includes a 'Compare Module' button and a '2/unlimited' status. A table lists users and their license usage. Callouts point to the 'Compare Module' button, the '2/unlimited' status, the 'License occupied by' column, and the 'Until' column.

Currently used licenses / available licenses

License occupied by

Until

Until: 19.9.2024, 09:36:40

Until: 20.9.2024, 09:46:16

Until: 19.9.2024, 09:36:40

Until: 20.9.2024, 09:46:16

requisis_Toolbox

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.

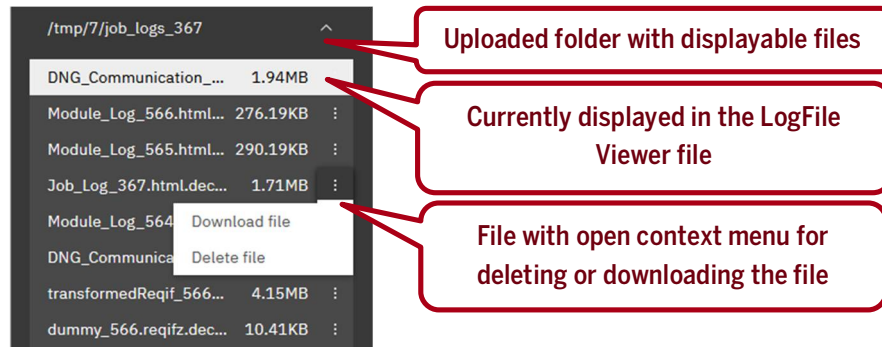
It is possible to filter by log level. You can see how many log entries there are in the currently selected file by the number of the respective filter. Click on the corresponding filter to select or deselect it. To customize the list of existing entries according to your wishes. If a file is selected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

The screenshot shows the LogFile Viewer interface with several annotations:

- deselected filter**: Points to the 'Trace' filter, which is unchecked.
- Selected filter**: Points to the 'Warning' filter, which is checked.
- Number of entries contained**: Points to the 'Warning 0' text next to the selected filter.
- Search function to find a specific entry**: Points to the search bar above the log table.
- Current content of the subfolder /app/var/logs for live analysis**: Points to the 'Files' sidebar on the right.
- Timestamp in the log.**: Points to the 'Time' column in the log table.
- Log level of the entry**: Points to the 'Level' column in the log table.
- log files within a zip file. selected file is displayed**: Points to a file entry in the 'Files' sidebar.
- upload log, txt or zip files**: Points to the 'Upload file' button at the bottom right.

To remove files from the list, click on the menu icon to the right of the corresponding file. There is currently no option to remove files from the *LogFile Viewer* folder by folder.

requisis_Toolbox



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).

requisis_Toolbox

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 Manager 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.

requisis_Toolbox

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.

requisis_Toolbox

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* 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* 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 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.DNG-ELM.com).
> > NEW-DOMAIN:	The new domain to be used in the future. (ex. new.IBM-ELM.com)

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

requisis_Toolbox

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



requisis_Toolbox

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:

`migrations-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

requisis_Toolbox


9 API

The *requisis ReqIF-Manager* has an API. The API functions can be studied best with a swagger editor like <https://editor.swagger.io/>.

Important: In order to use the API all `auth.ldap.binduser.*` parameters must be configured and enabled.

From version 2.0.1 on the API-Docummentation is also accessible using the following URI:

<http://<servername>/requisis/reqifmanager/apiDocummentation>

The About dialog can be found in the upper right corner of the GUI . There is a direct link to the API documentation.

9.1 Shared secret calculation

The API uses an API key which is generated as a time-based hash.

The client must build the hash and send it to the server. The shared secret consists of a public API key and salt build by timestamp that changes every 5 minutes.

The server checks this hash accordingly. Accepted are the keys of the current salt as well as the previous and following salt. Here an example how the hash of the shared secret is created:

PseudoCode:

```
salt = floor(current_unix_timestamp/300)    //timestamp for every 5 minutes
api-key = sha256(publicApiKey+salt)        //string concatenation
```

Example:

```
SharedSecret = 123456
The Current Unix Timestamp
1605600862 seconds since Jan 01 1970. (UTC)
salt = roundDown(5352002,8733..)
salt = 5352002
api-key = sha256(SharedSecret+salt)
api-key = sha256(1234565352002)
api-key = dd8a093c01dfef50d02eec4393ecdbf1da1cb27edc1f2dc4f2e602d7c1b10d21
```

Header information for API request:

```
"api-key: dd8a093c01dfef50d02eec4393ecdbf1da1cb27edc1f2dc4f2e602d7c1b10d21"
```

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>
```

requisis_Toolbox

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
```



requisis_Toolbox

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

```



requisis_Toolbox

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_MYSQL-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
```



requisis_Toolbox

11 Glossary

Concept	Meaning
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.
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.
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.
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 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.



requisis_Toolbox

Concept	Meaning
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 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-REQUIS, ext. 666

product-support@requisis.com, <https://requisis.com>