



/HydroMet

Deployment Guide

WISKI7 Administration and Configuration

 **KISTERS**
Empowering decisions of tomorrow

The software programs described in this document and the information contained in this document are confidential and proprietary products of KISTERS or its licensors. KISTERS waives copyright for licensed software users to print out parts of the documentation in hard copy for their own use only.

This documentation may not be transferred, disclosed, or otherwise provided to third parties. In duplicating any part of this document, the recipient agrees to make every reasonable effort to prevent the unauthorized use and distribution of the proprietary information.

No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

KISTERS reserves the right to make changes in specifications and other information contained in this publication without prior notice.

KISTERS makes no warranty of any kind with regard to this material including, but not limited to, the implied warranties or merchantability and fitness for a particular purpose.

KISTERS shall not be liable for any incidental, indirect, special or consequential damages whatsoever (including but not limited to lost profits) arising out of or related to this documentation, the information contained in it or from the use of programs and source code that may accompany it, even if KISTERS has been advised of the possibility of such damages.

Any errors found in any KISTERS product should be reported to KISTERS where every effort will be made to quickly resolve the problem.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

Table of Contents

1	Introduction	7
2	WISKI7 System Architecture	8
2.1	KiTSM Dispatcher	8
2.2	KiTSM Server	9
2.3	KiTSM Process Control and Administration (JProcController, WSM Service Monitor)	9
2.4	KiTSM Client Connection	9
3	WISKI7 Software Packages (Deliverables)	10
3.1	Package Naming Convention	10
3.2	Available Packages	10
3.3	Directory Structure Inside the Packages	11
3.4	Nice to Have	11
4	Installation Requirements	12
4.1	WISKI7 Server	12
4.1.1	Hardware and Software Requirements	12
4.1.2	Port Overview	14
4.2	WISKI7 Client	28
4.2.1	Hardware and Software Requirements	29
4.2.2	Installation Requirements	31
4.3	Databases and Storage	31
5	Preparations For a New Installation	33
5.1	Preparing the Database Server for the KISTERS WISKI Services	33
5.1.1	Installing Databases under ORACLE	33
5.1.2	Installing Databases under Microsoft SQL Server	34
5.1.3	Installing Databases under PostgreSQL	42
5.2	Preparing the WISKI Server: Setup the security settings for the installation directory	44
5.3	Preparing the WISKI Server: Install Perl	46
5.4	Preparing the Client Machine: ORACLE Client Setup	49
6	WISKI Server Manager (WSM)	50
6.1	Deployed WSM Packages	50
6.2	WSM User Interface	51
6.2.1	WSM Menus	51
6.2.2	WSM Tabs	52
6.3	Customise Actions	56
6.4	Log Viewer	60
7	Initial Installation Using the WSM (Empty Database)	61
8	Updating an Installation Using the WSM (Existing Database)	76
9	Duplicating KISTERS WISKI7 Server Database	89
9.1	Exporting Original Database	89

9.2	Configuring Connection Settings to WISKI7 Server Database Manually	92
9.2.1	Connection Entries for ORACLE	93
9.2.2	Connection Entries for Microsoft SQL Server	93
9.2.3	Set User Name and Password	93
9.3	Updating Dispatcher Database Connections	94
9.3.1	Checking and Correcting the TCA Database Storage	95
9.3.2	Registering the KiTSM Tenant Database Storage	96
9.3.3	Running the Complete Update Process	97
9.3.4	Checking the Background Process Configuration	98
9.3.5	Registering the New Dispatcher Database URL	99
9.4	Updating the Client Schema	99
10	WISKI Server Initial Installation (Manual Installation without WSM)	100
10.1	Manual Server Installation Example	100
10.1.1	Put together the packages which should be installed and extract the packages	100
10.1.2	Initialize the Server Database Part	100
10.1.3	Install the WISKI Services as a WINDOWS Service	101
10.1.4	Checking the KiTSM state	101
10.2	WISKI7 Client Installation Tasks	104
10.2.1	Initialising the WISKI7 Database	104
10.3	Post-installation Steps	112
11	WISKI Server Update (Manual Update)	113
11.1	Updating a previous installation with new directory structure	113
12	Setting Up a Satellite KiTSM Server	115
12.1	Setting Up a Calculation Node for Satellites	115
12.1.1	Required packages for Satellites	115
12.1.2	Since WISKI 7.4.3 SR12	115
12.1.3	Before WISKI 7.4.3 SR12	116
12.2	Setting Up an Interactive Node for Satellites	117
12.2.1	Required packages for an Interactive Node	117
12.2.2	Since WISKI 7.4.3 SR12	117
12.2.3	Before WISKI 7.4.3 SR12	119
13	Linux Installation	121
13.1	Initial Installation	121
13.2	Updating an Existing Installation	121
14	Certificate and TLS Management in WISKI	123
14.1	WISKI Server Configuration	125
14.2	WISKI Client Configuration	127
14.3	WISKI Portal Configuration	127
14.4	Security Monitor	127
15	Security	131
15.1	Authentication	131
15.1.1	Change Password for a WISKI Client User	131

15.1.2	Configuring LDAP Authentication	131
15.2	WISKI Server Security Tool	154
16	WISKI Troubleshooting	159
16.1	TLS Certificate Issues	159
16.2	Installation of Self-Signed Root CA in Local TrustStore	160
16.3	Database Problems	162
16.3.1	Repair Missing Database Constraints	163
16.4	Error message on WISKI startup "Connect to KWebKitObjCtrl failed -3"	163
16.5	Login Problems	163
16.5.1	No free TSM state	164
16.5.2	No open tenant MaiDomain or WISKI	164
16.5.3	Login failed for user 'wiskiadm'. while trying to get a pool for tenant 1001	164
16.5.4	TSM/TCA database connection troubleshooting: ODBC connection without DSN	164
16.5.5	TSM/TCA database connection troubleshooting: ODBC connection with DSN on 64-bit computer	165
16.5.6	Database tables missing or not up to date. Ask the administrator to update the database.	165
16.6	System Startup Problems	165
16.6.1	Dispatcher DB status is set to "Stop" (red) in the WSM Service Monitor	166
16.6.2	KiTSM Dispatcher status is set to "Stop" (red) in the WSM Service Monitor	166
16.6.3	Webservice status is set to "Stop" (red) in the WSM Service Monitor	166
16.6.4	TSM server status is set to "stop" (red) in the WSM Service Monitor	166
16.6.5	There are only "?" in the WSM Service Monitor instead of status indicators	166
16.7	Monitoring Your System	166
16.7.1	Using the JProcConsole	166
16.7.2	Using the Web Admin User Interface	168
16.8	Monitoring General KiTSM Server State	168
16.9	Monitoring KiTSM Calculation Queues	170
16.9.1	Adding a New Tenant	173
16.9.2	How to Determine and Control System Load	173
16.10	Solve performance issues on Windows 2022 with RDP connections	174
17	Optimising Your System	175
17.1	Managing Background Processes	175
17.2	Optimising Your System For Memory and CPU Usage	176
17.3	Optimising Background Processes/Queues	177
18	Using the WISKI Standard Portal	181
18.1	Portal Installation	181
18.2	Starting and Stopping	184
18.3	Portal Administration (since WISKI 7.4.11 SR7)	185
18.4	Troubleshooting for the Portal	186
18.5	Standard Applications	195
18.5.1	Application Metrics	195
18.5.2	Process Analytics	197
18.5.3	Data Viewer	199

18.5.4	KiDSM	200
18.5.5	TS Data Validation	203
19	Appendix	206
19.1	External Java installations	206
19.2	Windows Services	206
19.2.1	Managing a KiDSM Service As a WINDOWS service	206
19.2.2	Managing the KISTERS WISKI Service as a WINDOWS Service	207
19.3	Client Installation Using an Installation Wizard (WiX Toolset)	208
19.3.1	Installation	209
19.3.2	Deinstallation	217
19.3.3	Upgrade	219
19.4	Client Installation/Update in Silent Mode	220
19.4.1	Installation	220
19.4.2	Update	222
19.4.3	Deinstallation	223
19.5	Usage of Elasticsearch	223
19.5.1	Installation	224
19.5.2	Start Elasticsearch	226
19.5.3	Update Elasticsearch	226
19.5.4	Find Elasticsearch in the Service Registry	227
19.6	Python as Scripting Language	227
19.7	FAQ	228
19.7.1	FAQ on TLS and Certificate Management	228
19.7.2	FAQ on Keycloak	232
	Index	234

1 Introduction

This administrator documentation explains how to install, configure and update both the WISKI7 server and other modules such as KiIOSys, KiDSM, the Water Portal, etc.) as well as the WISKI7 client and the corresponding KISTERS services.

It gives an overview about the system architecture, the packages needed to make the server, the client and all required modules and services run as well as an overview about installation prerequisites and requirements.

You will find information on the WISKI Server Manager (WSM), including details of its various features and the Service Monitor, which is used to control the availability of the server. You will also find hints about general troubleshooting and optimising your system.

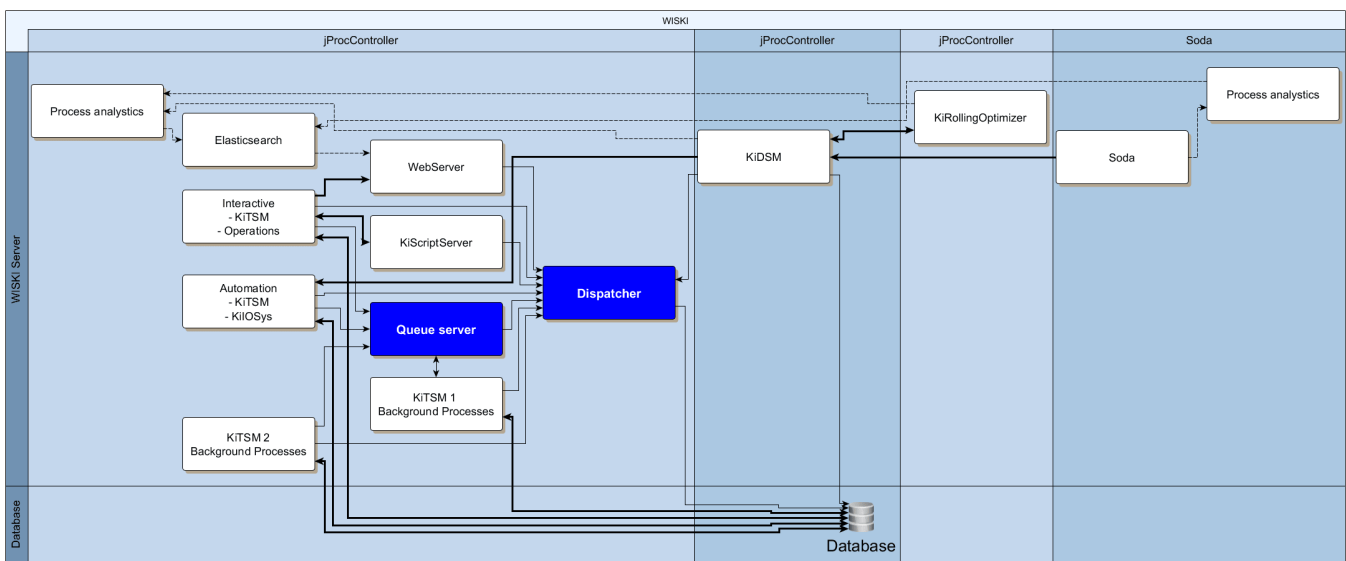
2 WISKI7 System Architecture

WISKI7 and the KISTERS Time Series Management System (KiTSM) is the result of almost 25 years of experiences collected in the global water industry. The requirements on data acquisition, storage, organisation, validation, analysis and integration and dissemination from the international market have been included in a reliable, scalable and controlled open multi-tier architecture.

KiTSM is the KISTERS time series management server developed in Java and is designed to organize, compute and share time series mass data.

KiTSM is the new shared system core of all KISTERS products where time series are involved. In three layer architecture, it represents the application layer, and provides all services necessary for time series management and calculation for applications built upon it.

The architecture of KiTSM is designed to operate as an autonomous server system, so that it may be adjusted to suit all manner of applications. This includes an independent administration component with the necessary user interfaces, a process check, active session management, and the actual time series management and calculation engine.



See the next subchapter(s) for more information:

- [KiTSM Dispatcher](#)  8
- [KiTSM Server](#)  9
- [KiTSM Process Control and Administration \(JProcController, WSM Service Monitor\)](#)  9
- [KiTSM Client Connection](#)  9

2.1 KiTSM Dispatcher

The KiTSM dispatcher is the central operating point of every KiTSM system. Every application requiring access to time series data or calculations must register a session with the KiTSM dispatcher. During this registration process, the KiTSM dispatcher authenticates the client and/or the user logged in to the client. Depending on how the KiTSM system is integrated into the application as a whole determines if it maintains its own user management or if it connects to a master user management system.

After authentication is successful, the KiTSM dispatcher will assign the client application with a KiTSM server, which is then responsible for this session and provides the actual services. This assignment is controlled through a generic load balancing algorithm; as an alternative, KiTSM servers may also be reserved for specific services. Each KiTSM dispatcher is able to control any amount of KiTSM servers.

As soon as a KiTSM server has been assigned to the requesting application, communication between the KiTSM server and TCA takes place directly over the KiTSM service interface.

2.2 KiTSM Server

The KiTSM Server provides the services necessary for time series management and calculations on time series data. The performance of the KiTSM System may be adjusted through successive parallelisation over multiple KiTSM Servers to meet the respective application requirements with regard to data volume, data throughput and calculation requirements. The internal management of the KiTSM System also guarantees the availability of the system as a whole, even if individual KiTSM Servers fail.

2.3 KiTSM Process Control and Administration (JProcController, WSM Service Monitor)

The JProcController component monitors all processes in the KiTSM System, and WSM Service Monitor is the associated interface for the system administrator. The WSM Service Monitor is a client program that can be installed anywhere. From there, you can monitor any server process. Processes may be started and stopped through this interface, and additional KiTSM Servers added, for example, if calculation demand increases.

Each KiTSM System also offers a web-based administration console (TsmSAdmin) for additional administration and configuration tasks. This also facilitates remote maintenance of the entire system.

2.4 KiTSM Client Connection

TSM Client Applications (TCAs) connect through the KiTSM service interface, which is implemented in a so-called TCA Proxy. This is an object-oriented programming interface (either Java or C# / .NET) which can be integrated into client applications and transparently provides all executable time series management functions in the KiTSM. The TCA Proxy is based on the KiRPC (KISTERS Remote Procedure Call) communication mechanism, which is specially optimized for processing and transfer of time series data, and also supports mass data streaming in addition to the purely functional interfaces.

Any amount of KiTSM sessions may be active on a single client, and these may well be handled by different KiTSM servers. An integrated session manager, together with the KiTSM Dispatcher, ensures that an active connection between the TCA and a KiTSM Server always automatically becomes available.

3 WISKI7 Software Packages (Deliverables)

WISKI is supplied in packages. Customer packages are provided by KISTERS.

Detailed information can be found in the next subchapters:

- [Package Naming Convention](#)^[10]
- [Available Packages](#)^[10]
- [Directory Structure Inside the Packages](#)^[11]
- [Nice to Have](#)^[11]

3.1 Package Naming Convention

All WISKI packages from version WISKI 7.3 onward follow a naming convention:

WISKI-<version number>-<service release>-<package name>-<platform>.zip

Note:

- <platform> is optional.
- <package name> may contain one or multiple hyphens.

Examples:

- WISKI-7.4.11-SR14-kidsm.zip
- WISKI-7.4.11-SR14-wiski-client.zip

3.2 Available Packages

To decide which package you need in order to install your desired WISKI7 system, you need to know which modules you need and which operating system and platform you use. See [Package selection](#)^[65].

The following list shows some examples of installation packages. You can view the complete list of packages available to you when installing or updating via the WISKI Server Manager in the installation step **PACKAGE SELECTION**.

No.	Name	Part of the default installation	Extra license required	WINDOWS	LINUX	Remarks/Contents
1	WISKI-<vn>-<sr>-java-64bit.zip	X		X		Java (C) runtime environment
2	WISKI-<vn>-<sr>-java-64bit-linux.tar.gz	X			X	Java (C) runtime environment
3	WISKI-<vn>-<sr>-kidsm.zip		X	X	X	KiDSM server part
4	WISKI-<vn>-<sr>-kidsm-client-64.zip		X	X		KiDSM client

Note:

<vn> = <version number>

<sr> = <service release>

3.3 Directory Structure Inside the Packages

Select a root directory and unpack the zip package there. The package is unzipped under its top level folder named **\WISKI**.

Directory	Description
<root directory>\wiski	Root directory of the WISKI software installation.
<root directory>\wiski\bin	Contains all scripts needed to maintain the installed software.
<root directory>\wiski\clients	Contains all GUI clients; for each client a separate sub folder is created (e.g. C:\kisters\wiski\clients\wiski-client).
<root directory>\wiski\info	Contains text files with version information about the installed packages. Folder is only created if you use the respective package.
<root directory>\wiski\services	Contains the server software, including all components (WISKI server, KiDSM, KiIOSys, etc.). Folder is only created if you use the respective package.
<root directory>\wiski\tools	Contains additional KISTERS tools.
<root directory>\documents <root directory>\documents\kidsm\damaged <root directory>\documents\kidsm\input <root directory>\documents\kidsm\output <root directory>\documents\kidsm\processing	Can be used to store data files such as incoming data for a hot folder. The \documents folder is only included in the KiDSM package.

3.4 Nice to Have

To make your work with WISKI7 easier, use these tools.

Zipper for tarballs, gz, zip, rar files	https://www.7-zip.de/
Dependency Walker, detects missing DLLs	https://www.dependencywalker.com/
Textfile editor	https://notepad-plus-plus.org/
FTP client	https://filezilla-project.org/

4 Installation Requirements

This administrator's manual explains how to set up the WISKI7 server and client and all required modules.

As WISKI7 closely interacts with the database, make sure that both the server components (KiTSM, KiDSM, KiALM (the AlarmManager) and the client computers have a good network connection to the database server.

First the hardware and software requirements for the installation are described. They are related to these topics:

- [WISKI7 Server](#) 
- [WISKI7 Client](#) 
- [Databases and Storage](#) 

4.1 WISKI7 Server

The requirements for a complete WISKI server installation comprise these topics:

- [Hardware and Software Requirements](#) 
- [Port Overview](#) 

4.1.1 Hardware and Software Requirements

Hardware

The hardware requirements depend on several parameters, such as the number of users, the amount of data to be processed (imports and exports), the complexity of the calculations, etc.

In general, a WISKI7 / WISKI Server installation supports virtualisation, but it is not recommended by KISTERS. Virtual hardware shares hardware resources with other virtual machines, which can lead to unexpected system behaviour.

Especially when processing raster data, an increased memory requirement of more than 32 GB can be expected.

	Minimum	Recommended
RAM	32 GB	64 GB
Disk Space	100 GB free (excluding the operating system)	500 GB free (excluding the operating system)
Network Bandwidth	Bandwidth LAN 100/1000 MBit/s	Bandwidth LAN 100/1000 MBit/s
CPU Cores	4	> 6

Example: MSSQL database server with a WISKI7 Server installation

- 2 × Intel® Xeon® Gold 6136 3.0G,12C/24T,10.4 GT/s 2UPI,24.75M Cache,Turbo,HT (150 W) DDR4-2666
- 6 × 32 GB RDIMM, 2666 MT/s, Dual Rank
- PERC H740P RAID Controller, 8Gb NV Cache, Adapter, Low Profile
- RAID 10 for HDDs or SSDs in pairs (Matching Type/Speed/Capacity)
- 10 × 480 GB SSD SATA Mix Use MLC 6Gbps 512n 2.5 in Hot-plug Drive, SM863a
- QLogic 57800 2 × 10Gb BT + 2 × 1Gb BT Network Daughter Card
- Dual, Hot-plug, Redundant Power Supply (1+1), 750 W, Titanium, 200 - 240 V AC
- iDRAC9 Enterprise with OpenManage Essentials, Server Configuration Management
- PowerEdge 2U LCD Bezel
- ReadyRails™ Sliding Rails With Cable Management Arm
- Electronic System Documentation and OpenManage DVD Kit
- MS-Windows Server® 2022,Standard Ed,16CORE,Factory Inst,2VM,NO CAL, Media Kit
- MS-Windows Server® 2022,Standard Ed, Add License,16CORE,NO MEDIA/KEY
- 5Yr ProSupport and 4hr Mission Critical

Other Requirements

Supported operating systems	Windows 2019, Windows 2022, Windows 2025 (from WISKI version 7.4.13), Linux, Unix (CentOS is recommended) Note: Only the server components can be installed on a Linux / Unix system.
Privileges at installation time	Full access, rights to create / change / delete databases. The Database must be created with a case sensitive Encoding (e.g. German_PhoneBook_BIN or Latin1_General_CS_AS).
Privileges at runtime	Starting and stopping of services, writing/creating log and data files. Network communication (see Port Overview ^[14]) Full access, rights to create / change / delete database tables.
Installation folder	Any folder name without blanks, no network path, no system protected paths for the server part (c:\program files). Read and write access to the installation directory must be granted to all users who will be performing installations. Full control of the directories is not required. The client can be installed under c:\program files .
Full DNS support needed!	For a stable WISKI Server operation DNS must know all workstations in the network that communicate with the WISKI Server. DNS must be capable to resolve name requests by the workstations' IP addresses because within Firewalls and virtual machines a Network Address Translation (NAT) is executed. Besides, DHCP could lead to address conflicts, if the WISKI Server is configured to use IP addresses.
Direct network connections	WISKI Server currently does not support the use of HTTP proxies. Therefore, all components of the server must be connected via direct connections without using HTTP proxies. As of 09-2024, load balancer configurations are no longer supported.
Host name	Note that the host names must fulfil the following conditions: <ul style="list-style-type: none"> ▪ The server requires a static host name. ▪ The host name must not contain a '_'.
Connection to the Database (ORACLE)	Connection through JDBC client (default Port 1521)
Connection to the Database (MSSQL)	Connection through JDBC client (default Port 1433)
PostgreSQL	Connection through JDBC client (default Port 5432)
Valid database password characters	Only the following ANSI characters are valid for database passwords: a-z, A-Z, 0-9, @!^[]{}<>_*.,:~ Note: Even if the database supports more characters, other characters are not supported by WISKI.
Connection to WISKI7 Client	Connection to Dispatcher/WISKI Server
Third Party	Perl and Java SDK (part of delivery) For Java SDK see information in the appendix ^[206] .

4.1.2 Port Overview

Default Ports (partially configurable)

Note: In order to run a WISKI7 on a client machine the following ports have to be accessible.

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
Automation	WISKI-<vn>-<sr>-services.zip	KisRPC	KisRPC	Java	7440, 7437 (pre-7.4.11-SR10 installations: 7442-7459*)	Automation	x	x		x
Automation	WISKI-<vn>-<sr>-services.zip	Metrics	HTTP	Java	7401	Automation		x	x	x
Automation	WISKI-<vn>-<sr>-services.zip	Metrics	HTTPS	Java	7418	Automation		x	x	x
Automation	WISKI-<vn>-<sr>-services.zip	Shutdown		Java	7490-7499*	Automation				
Dispatcher	WISKI-<vn>-<sr>-services.zip	Tsm Admin Interface	HTTP	Java	7420	Dispatcher	x	x		
Dispatcher	WISKI-<vn>-<sr>-services.zip	Tsm Admin Interface	HTTPS	Java	7421	Dispatcher	x	x		
Dispatcher	WISKI-<vn>-<sr>-services.zip	KisRPC	KisRPC	Java	7430	Dispatcher	x	x		
Dispatcher	WISKI-<vn>-<sr>-services.zip	Metrics	HTTP	Java	7405	Dispatcher		x	x	
Dispatcher	WISKI-<vn>-<sr>-services.zip	Metrics	HTTPS	Java	7410	Dispatcher		x	x	
Dispatcher	WISKI-<vn>-<sr>-services.zip	Shutdown		Java	7490-7499*	Dispatcher				

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
ElasticSearch	WISKI-<vn>-<sr>-elasticsearch.zip		HTTP	Java	9200	Elasticsearch			x	
Interactive	WISKI-<vn>-<sr>-services.zip	KisRPC	KisRPC	Java	7433, 7434, 7435 (pre-7.4.11-SR10 installations: 7442-7459*)	Interactive	x	x		x
Interactive	WISKI-<vn>-<sr>-services.zip	Metrics	HTTP	Java	7406	Interactive		x	x	x
Interactive	WISKI-<vn>-<sr>-services.zip	Metrics	HTTPS	Java	7408	Interactive		x	x	x
Interactive	WISKI-<vn>-<sr>-services.zip	WISKI REST API	HTTP	Java	7406	Interactive	x	x		x
Interactive	WISKI-<vn>-<sr>-services.zip	WISKI REST API	HTTPS	Java	7408	Interactive	x	x		x
Interactive	WISKI-<vn>-<sr>-services.zip	Shutdown		Java	7490-7499*	Interactive				
JProcController KiDSM	WISKI-<vn>-<sr>-kidsm.zip	RMI	RMI	Java	8999	JProcController		x	x	
JProcController KiDSM	WISKI-<vn>-<sr>-kidsm.zip	Shutdown		Java	8998	JProcController		x	x	
JProcController KiDSM	WISKI-<vn>-<sr>-kidsm.zip	Service port	RMI	Java	7489	JProcController				
JProcController KiDSM	WISKI-<vn>-<sr>-kidsm.zip	JProcController REST API	HTTPS	Java	7426	JProcController		x	x	
JProcController KiRo	WISKI-<vn>-<sr>-kirollingoptimizer.tar.gz	RMI	RMI	Java	7999	JProcController				

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
JProcController KiRo	WISKI-<vn>-<sr>-kirollingoptimizer.tar.gz	Shutdown		Java	7998	JProcController				
JProcController KiRo	WISKI-<vn>-<sr>-kirollingoptimizer.tar.gz	Service port	RMI	Java	8000	JProcController				
JProcController KiRo	WISKI-<vn>-<sr>-kirollingoptimizer.tar.gz	JProcController REST API	HTTPS	Java	7427	JProcController		x	x	
JProcController WISKI	WISKI-<vn>-<sr>-jproc.tar.gz	RMI	RMI	Java	1099	JProcController		x		
JProcController WISKI	WISKI-<vn>-<sr>-jproc.tar.gz	Shutdown		Java	7488	JProcController				
JProcController WISKI	WISKI-<vn>-<sr>-jproc.tar.gz	Service port	RMI	Java	7500	JProcController				
JProcController WISKI	WISKI-<vn>-<sr>-jproc.tar.gz	JProcController REST API	HTTPS	Java	7425	JProcController		x	x	
Keycloak	WISKI-<vn>-<sr>-keycloak.tar.gz	REST API	HTTP		7414			x		
Keycloak	WISKI-<vn>-<sr>-keycloak.tar.gz	Secure REST API	HTTPS		7415		x	x		
Keycloak	WISKI-<vn>-<sr>-keycloak.tar.gz	Management Port	HTTPS		7416			x		
KiDSM	WISKI-<vn>-<sr>-kidsm.zip	ActiveMQ Broker	Open Wire	Java	8800	KiDSM internal			x	
KiDSM	WISKI-<vn>-<sr>-kidsm.zip	KiDSM Portal Application	HTTPS	Java	8843	KiDSM	x	x	x	
KiDSM	WISKI-<vn>-<sr>-kidsm.zip	KiDSM RCP	RCP	Java	8888-8889	KiDSM Client RCP	x	x	x	

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
		Client (until 7.4.7)								
KiDSM	WISKI-<vn>-<sr>-kidsm.zip	internal REST API		Java	8881	KiDSM internal		x	x	
KiDSM	WISKI-<vn>-<sr>-kidsm.zip	JProcCon troller Port Connecto r Server		Java	8901	KiDSM internal		x	x	
KiDSM	WISKI-<vn>-<sr>-kidsm.zip	JProcCon troller Port Time Scheduler		Java	8902	KiDSM internal		x	x	
KiDSM	WISKI-<vn>-<sr>-kidsm.zip	JProcCon troller Port Task Manager		Java	8903	KiDSM internal		x	x	
KiDSM	WISKI-<vn>-<sr>-kidsm.zip	JProcCon troller Port Transfer Tools		Java	8904	KiDSM internal		x	x	
KiDSM	WISKI-<vn>-<sr>-kidsm.zip	JProcCon troller Port Program Executor		Java	8905	KiDSM internal		x	x	
KiRollingOptimi zer	WISKI-<vn>-<sr>-kirollingoptimizer.tar.gz	ActiveMQ Broker	Open Wire	Java	1883	KiRollingOpti mizer internal			x	
Log Server	No longer part of current versions			Java	8301	Log Server				
Log Server	No longer part of current versions			Java	8302	Log Server				

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
Microsoft SQL Server	---				1433	Microsoft SQL Server			x	
nth cluster application	WISKI- $\langle vn \rangle$ - $\langle sr \rangle$ -services.zip	API	KisRPC		7441 + n					
nth dcserver instance	WISKI- $\langle vn \rangle$ - $\langle sr \rangle$ -services.zip	Shutdown	Controller		7489 + n					
Oracle Server	---				1521	Oracle Server			x	
os-monitor	No longer part of current versions	Shutdown	Controller		7473					
Python-FastAPI-Runner	WISKI- $\langle vn \rangle$ - $\langle sr \rangle$ -water-python-win64.zip WISKI- $\langle vn \rangle$ - $\langle sr \rangle$ -python-fastapi-jproc.tar.gz	Water Python FastAPI components runner	HTTPS	Java	7466	Tomcat	x	x	x	x
Python-FastAPI-Runner	WISKI- $\langle vn \rangle$ - $\langle sr \rangle$ -water-python-win64.zip WISKI- $\langle vn \rangle$ - $\langle sr \rangle$ -python-fastapi-jproc.tar.gz	Water Python FastAPI components runner	HTTP	Python	7467	Python-FastAPI-Runner		x	x	x
Queueserver	WISKI- $\langle vn \rangle$ - $\langle sr \rangle$ -services.zip	KisRPC	KisRPC	Java	7436 (pre-7.4.11-SR10 installations: 7442-7459*)	Queueserver	x	x		
Queueserver	WISKI- $\langle vn \rangle$ - $\langle sr \rangle$ -services.zip	Metrics	HTTP	Java	7402	Queueserver		x	x	
Queueserver	WISKI- $\langle vn \rangle$ - $\langle sr \rangle$ -services.zip	Metrics	HTTPS	Java	7419	Queueserver		x	x	

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
Queueserver	WISKI-<vn>-<sr>-services.zip	Shutdown		Java	7490-7499*	Queueserver				
Queueserver	WISKI-<vn>-<sr>-services.zip	Lock service	RIM	Java	7429	Queueserver		x		x
Raster Extractions	WISKI-<vn>-<sr>-raster-extractions.tar.gz	External Raster Extractions	HTTP	Java	7474	Raster-Extractions			x	x
Raster Extractions	WISKI-<vn>-<sr>-raster-extractions.tar.gz	External Raster Extractions	HTTPS	Java	7475	Raster-Extractions			x	x
Raster Extractions	WISKI-<vn>-<sr>-raster-extractions.tar.gz	External Raster Extractions	SSH	Java	7476	Raster-Extractions				
Raster Extractions	WISKI-<vn>-<sr>-raster-extractions.tar.gz	External Raster Extractions	RMIRegistry Port	Java	7477	Raster-Extractions				
Raster Extractions	WISKI-<vn>-<sr>-raster-extractions.tar.gz	External Raster Extractions	RMIServerPort	Java	7478	Raster-Extractions				
Scriptserver	WISKI-<vn>-<sr>-services.zip	KisRPC	KisRPC	Java	7438, 7439 (pre-7.4.11-SR10 installations: 7442-7459*)	Scriptserver	x	x		x
Scriptserver	WISKI-<vn>-<sr>-services.zip	Scriptserver Secure	Scriptserver	Java	7432	Scriptserver	x	x		x

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
Scriptserver	WISKI-<vn>-<sr>-services.zip	Metrics	HTTP	Java	7403	Scriptserver		x	x	x
Scriptserver	WISKI-<vn>-<sr>-services.zip	Metrics	HTTPS	Java	7422	Scriptserver		x	x	x
Scriptserver	WISKI-<vn>-<sr>-services.zip	Shutdown		Java	7490-7499*	Scriptserver				
Security-Monitor	WISKI-<vn>-<sr>-security-distribution.tar.gz	Process that monitors e.g. TLS certificate status	HTTP (localhost only)	Java	7411	Security-Monitor				x
Service Node 0	WISKI-<vn>-<sr>-services.zip	RMI	RMI	Java	999	Service Node 0		x		
Service Node 0 ssh	WISKI-<vn>-<sr>-services.zip	running on 127.0.0.1 only	RMI	Java	7412	Service Node 0			x	x
Service Node 0	WISKI-<vn>-<sr>-services.zip	RMI	RMI	Java	5444	Service Node 0				
Service Node 0	WISKI-<vn>-<sr>-services.zip	Web Server REST API	HTTP	Java	7480	Service Node 0	x	x		
Service Node 0	WISKI-<vn>-<sr>-services.zip	Web Server REST API	HTTPS	Java	7479	Service Node 0	x	x		
Time Series Data (TSDI)	WISKI-<vn>-<sr>-tsdi.zip	Time series data import (changed from 7.4.13 SR 11)	RMI Registry	Java	7471	Time Series Data Import				

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
Time Data (TSDI) Series Import	WISKI-<vn>-<sr>-tsdi.zip	Time series data import (changed from 7.4.13 SR 11)	RMI Server	Java	7472	Time Series Data Import				
Time Data (TSDI) Series Import	WISKI-<vn>-<sr>-tsdi.zip	Time series data import (changed from 7.4.13 SR 11)	SSH	Java	7484	Time Series Data Import				
Time Data (TSDI) Series Import	WISKI-<vn>-<sr>-tsdi.zip	Time series data import (changed from 7.4.13 SR 11)	HTTP	Java	7485	Time Series Data Import	x			x
Time Data (TSDI) Series Import	WISKI-<vn>-<sr>-tsdi.zip	Time series data import (changed from 7.4.13 SR 11)	HTTPS	Java	7486	Time Series Data Import	x			x
Tomcat	WISKI-<vn>-<sr>-tomcat.tar.gz	Portal Applications (incl. KiWIS)	HTTP	Java	7481	Tomcat	x	x		
Tomcat	WISKI-<vn>-<sr>-tomcat.tar.gz	Portal Applications	HTTPS	Java	7482	Tomcat	x	x		

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
		(incl. KiWIS)								
Tomcat	WISKI-<vn>-<sr>-tomcat.tar.gz	Shutdown		Java	7483	Tomcat				
Tsm1	WISKI-<vn>-<sr>-services.zip	KisRPC	KisRPC	Java	7441 (pre-7.4.11-SR10 installations: 7442-7459*)	Tsm1	x	x		x
Tsm1	WISKI-<vn>-<sr>-services.zip	Metrics	HTTP	Java	7404	Tsm1		x	x	x
Tsm1	WISKI-<vn>-<sr>-services.zip	Metrics	HTTPS	Java	7423	Tsm1		x	x	x
Tsm1	WISKI-<vn>-<sr>-services.zip	Shutdown		Java	7490-7499*	Tsm1				
Validation Karaf (Executor, Persistence, Queue)	WISKI-<vn>-<sr>-validation-karaf.tar.gz	SSH	SSH	Java	7461	Validation-Karaf	x		x	x
Validation Karaf (Executor, Persistence, Queue)	WISKI-<vn>-<sr>-validation-karaf.tar.gz	RMI Registry	RMI	Java	7462	Validation-Karaf	x		x	x
Validation Karaf (Executor, Persistence, Queue)	WISKI-<vn>-<sr>-validation-karaf.tar.gz	RMI Server	RMI	Java	7463	Validation-Karaf	x		x	x
Validation Karaf (Executor, Persistence, Queue)	WISKI-<vn>-<sr>-validation-karaf.tar.gz	Validation Karaf REST services	HTTPS	Java	7464	Validation-Karaf	x		x	x
Validation Karaf (Executor, Persistence, Queue)	WISKI-<vn>-<sr>-validation-karaf.tar.gz	Validation Karaf	HTTP	Java	7465	Validation-Karaf	x		x	x

Process	Deployment Package	Description	Protocol	Executable	Port	Process Help	Open in Firewall (Desktop Client Access)	Automatic Port Offset	Optional Service	Available for Satellite Nodes
Persistence, Queue)	karaf.tar.gz	REST services								
Water-Python-Components	WISKI-<vn>-<sr>-water-python-win64.zip WISKI-<vn>-<sr>-python-jproc.tar.gz	Water Python component framework	HTTPS	Java	7468	Tomcat	x	x	x	x
Water-Python-Components	WISKI-<vn>-<sr>-water-python-win64.zip WISKI-<vn>-<sr>-python-jproc.tar.gz	Water Python component framework	HTTP	Python	7469	Water-Python-Components		x	x	x
WISKI-SingleNode	WISKI-<vn>-<sr>-services.zip Only with SingleNode profile (not for production use)	WISKI REST API	HTTP	Java	7407	SingleNode	x	x		x
WISKI-SingleNode	WISKI-<vn>-<sr>-services.zip Only with SingleNode profile (not for production use)	WISKI REST API	HTTPS	Java	7409	SingleNode	x	x		x
Workflow Engine	WISKI-<vn>-<sr>-water-workflow-engine.zip	Workflow engine used by data validation and for the WISKI workflows	HTTP	Java	7470	Workflow Engine	x		x	x

*Cluster nodes, the ports are automatically applied on initial installation and can be changed in the KiTSM Webinterface at KiTSM-Cluster/Cluster-Nodes (<http://localhost:7420/TsmSAdmin/tsmservers.do>).

Note: Any port within the range 7442-7459 can still be used in the current installation if the initial installation was carried out with WISKI version 7.4.11-SR10 or later.

Home TSM Cluster Log Viewer Tenants Authorisation Preferences											
Controller Cluster Nodes Cluster Load Dispatcher Cluster Sessions Background Processes											
<div> <div>1 - 5 / 5</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div>											
Key	Instance Id	Host	Instance Name	Service Name	State	Adm State	Heartbeat Date	Tolerance [min]	Load	URL	Software
103001	13939001	10.10.10.10	10	queueserver	on	on	16.10.2018 10:10:54 CEST	2	0.00	rmi://10.10.10.10:1099/ClusterNodeControl10	development system 27.09.2018 14:55:39 CEST
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
102001	tsm-queue-server	tsm-queue-server	7442	on	on	0 (0)	0		true		
203001	13943001	10.10.10.10	50	scriptserver	on	on	16.10.2018 10:11:06 CEST	2	0.96	rmi://10.10.10.10:1099/ClusterNodeControl50	development system 27.09.2018 14:55:39 CEST
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
204001	scriptserver	scriptserver	7444	on	on	1 (0)	1		true		
206001	wiski-control	WISKI-control	7446	on	on	1 (0)	0		true		
204001	13941001	10.10.10.10	40	interactive	on	on	16.10.2018 10:11:00 CEST	2	0.05	rmi://10.10.10.10:1099/ClusterNodeControl40	development system 27.09.2018 14:55:39 CEST
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
205001	operationserver	OperationServer	7445	on	on	3 (0)	0		true		
209001	tsmservice	tsmservice	7449	on	on	11 (0)	0		clientGroup != "kiosys"		
212001	wiski-control	WISKI-control	7452	on	on	1 (0)	0		true		
213001	wiski-restapi	WISKI-RESTAPI	7453	on	on	0 (0)	0		true		
214001	wiskiserver	WISKIServer	7454	on	on	5 (0)	0		true		
205001	13942001	10.10.10.10	30	tsm1	on	on	16.10.2018 10:11:07 CEST	2	0.00	rmi://10.10.10.10:1099/ClusterNodeControl30	development system 27.09.2018 14:55:39 CEST
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
208001	tsmservice	tsmservice	7448	on	off	2 (2)	0		true		
210001	wiski-control	WISKI-control	7450	on	on	1 (0)	0		true		
206001	13940001	10.10.10.10	20	automation	on	on	16.10.2018 10:10:53 CEST	2	0.00	rmi://10.10.10.10:1099/ClusterNodeControl20	development system 27.09.2018 14:55:39 CEST
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
203001	kiosys	KIOSSystem	7443	on	on	0 (0)	0		true		
207001	tsmservice	tsmservice	7447	on	on	0 (0)	0		clientGroup == "kiosys"		
211001	wiski-control	WISKI-control	7451	on	on	1 (0)	0		true		

Firewalls and RMI Ports

When using a firewall, some of the RMI server sockets must be set to fixed values so that the firewall configuration can be configured to allow network communication through these sockets. The TCP connections between the TSM-Dispatcher, the TSM-Queueserver and the cluster node applications must be allowed by the firewall settings.

Dispatcher

Configure the RMI server ports of the TSM-Dispatcher by changing `configuration/dscmgr-dispatcher-custom.properties` and set the properties

- `tsms.dispatcher.rmi.serviceport` and
- `tsms.dispatcher.rmi.adminserviceport`

to a value greater than zero. In the firewall settings, allow TCP connections on these ports in both directions.

Cluster Node Controller

Configure the RMI server port of the cluster node application by changing `configuration/dscmgr-cluster-nodecontroller-custom.properties` and set the property

- `cluster.node.rmi.serviceport`

to a value greater than zero. In the firewall settings, allow TCP connections on this port in both directions. Do not confuse the setting with the local RMI registry port property `cluster.node.rmi.port`.

Queue Server

Note: This configuration is mandatory since TSM 3.66.44 or higher if running behind a firewall, but has no effect in older systems.

Configure the TSM Lock RMI service port by changing `dscmgr-kitsm-queueserver-custom.properties` and set the properties

- `lockComponent.pid=de.kisters.service.water.tsm.tslock.impl.osgi.ComponentConfiguration` and
- `lockComponent.serviceport=7429`

Use a value of **7429**. In the firewall settings, allow TCP connections on this port in both directions. Connections are established by TSM-Dispatcher and the TSM-Server nodes.

JProcController (optional)

If there is a need to connect a WSM Service Monitor from another computer to a computer running behind a firewall, you need to fix the RMI server port of JProcController by changing the `configuration/jproccontroller-custom.properties` and setting the property

- `jproccontroller.rmi.serviceport`

to a value greater than zero. In the firewall settings, allow TCP connections on this port in both directions.

Ports Communication Matrix

To / From	WISKI Desktop	Web Browser	JProcController	Dispatcher	Queueserver	Automation	Scriptserver	Interactive	Tsm1	Service Node 0	Tomcat Portal Applications	Workflow Engine	Execution Queue	KIDSM	Validation Persistence	Validation Executor	Water-Python-Components	Raster - Extraction	Security-Monitor
JProcController RMI																			
JProcController Shutdown																			
Dispatcher Tsm Admin Interface	74 20	74 20								74 20	74 20	74 20	74 20		74 20	74 20	74 20	74 20	
Dispatcher Tsm Admin Interface	74 21	74 21								74 21	74 21	74 21	74 21		74 21	74 21	74 21	74 21	
Dispatcher KisRPC	74 30				74 30	74 30	74 30		74 30	74 30	74 30								
Dispatcher Metrics																			
Dispatcher Shutdown			74 90- 74 99*																
Queueserver KisRPC				74 42- 74 59*		74 42- 74 59*	74 42- 74 59*		74 42- 74 59*										
Queueserver Metrics																			
Queueserver Shutdown			74 90- 74 99*																
Scriptserver KisRPC	74 42- 74 59*			74 42- 74 59*						74 42- 74 59*	74 42- 74 59*								
Scriptserver Metrics																			

To / From	WISKI Desktop	Web Browser	JProcController	Dispatcher	Queueserver	Automation	Scriptserver	Interactive	Tsm1	Service Node 0	Tomcat Portal Applications	Workflow Engine	Execution Queue	KIDSM	Validation Persistence	Validation Executor	Water-Python-Components	Raster - Extraction	Security-Monitor
Scriptserver Shutdown			74 90- 74 99*																
Interactive KisRPC	74 42- 74 59*			74 42- 74 59*		74 42- 74 59*	74 42- 74 59*			74 42- 74 59*	74 42- 74 59*								
Interactive Metrics																			
Interactive WISKI REST API	74 42- 74 59*						74 42- 74 59*			74 42- 74 59*	74 42- 74 59*						74 42- 74 59*	74 42- 74 59*	
Interactive Shutdown			74 90- 74 99*																
Tsm1 KisRPC	74 42- 74 59*			74 42- 74 59*	74 42- 74 59*														
Tsm1 Metrics																			
Tsm1 Shutdown			74 90- 74 99*																
Automation KisRPC	74 42- 74 59*			74 42- 74 59*			74 42- 74 59*			74 42- 74 59*	74 42- 74 59*								
Automation Metrics																			
Automation Shutdown			74 90- 74 99*																

To / From	WISKI Desktop	Web Browser	JProcController	Dispatcher	Queueserver	Automation	Scriptserver	Interactive	Tsm1	Service Node 0	Tomcat Portal Applications	Workflow Engine	Execution Queue	KIDSM	Validation Persistence	Validation Executor	Water-Python-Components	Raster - Extraction	Security-Monitor
Service Node 0 RMI																			
Service Node 0 RMI																			
Service Node 0 Web Server	74 80	74 80									74 80	74 80	74 80		74 80	74 80	74 80		
Service Node 0 Event Admin																			
Service Node 0 Time series data event																			
Tomcat Portal Applications	74 81	74 81																	
Workflow Engine											74 70								
Execution Queue																			
Validation Persistence											74 63								
Validation Executor																			
Water-Python-Components	74 69	74 69								74 69	74 69				74 69	74 69			
ElasticSearch										92 00	92 00								
Log Server																			
Log Server																			
Oracle Server	15 21		15 21	15 21	15 21	15 21	15 21		15 21	15 21	15 21	15 21							
Microsoft Server SQL	14 33		14 33	14 33	14 33	14 33	14 33		14 33	14 33	14 33	14 33							

To / From	WISKI Desktop	Web Browser	JProcController	Dispatcher	Queueserver	Automation	Scriptserver	Interactive	Tsm1	Service Node 0	Tomcat Portal Applications	Workflow Engine	Execution Queue	KIDSM	Validation Persistence	Validation Executor	Water-Python-Components	Raster - Extraction	Security-Monitor
Time Series Data Import (TSDI)	74 85 74 86													74 85 74 86					
Raster - Extraction						74 74		74 74	74 74										
Security-Monitor																			

*Cluster nodes, the ports are automatically applied on initial installation and can be changed in the KiTSM Webinterface at KiTSM-Cluster/Cluster-Nodes (<http://localhost:7420/TsmSAdmin/tsmservers.do>).

Note: Any port within the range 7442-7459 can still be used in the current installation if the initial installation was carried out with WISKI version 7.4.11-SR10 or later.

Check the ports by typing **netstat** at the command line to ensure that the WISKI7 server can use the required ports. We recommend that you use port offsets (such as 50,000) to be able to run more than one server.

4.2 WISKI7 Client

The requirements for a complete WISKI client installation comprise these topics:

- [Hardware and Software Requirements](#) ²⁹
- [Installation Requirements](#) ³¹

4.2.1 Hardware and Software Requirements

Requirements Concerning Hardware

Single Computer Installation

	Minimum (test system)	Recommended
RAM	16 GB	32 GB
Disc space	20 GB free (excluding the operating system)	20 GB free (excluding the operating system)
Network bandwidth	Bandwidth LAN 100 MBit/s	Bandwidth LAN 100/1000 MBit/s
Screen resolution	1920 × 1080	1920 × 1080
Screen font size	100 % - 200 %	100 % - 150 %
	A default printer must be available.	

Hardware for Terminal Server Installation

Note: Depending on the number of concurrent users, the requirements must be adapted.

	Recommended
RAM	32 GB
Disc space	20 GB free (excluding the operating system)
Network bandwidth	Bandwidth LAN 100/1000 MBit/s
Screen resolution	1920 × 1080
	A default printer must be available.

Note: Virtual machines and Citrix are supported; the hardware requirements have to be defined individually depending on the customer needs.

Requirements Concerning Software

Supported operating systems	MS Windows 2019, Windows 2022, Windows 2025 (from WISKI version 7.4.13), Windows 10, Windows 11; all with the latest Service Pack Note: <ul style="list-style-type: none">▪ If the WISKI7 client is installed on a 64-bit operating system, you must install the 32-bit database client to access the ORACLE TCA database.▪ For WISKI 7.4.11, the Microsoft Media Feature Pack must be installed (server management: activate MEDIA FOUNDATION).
Privileges installation time	at Administrator rights allowing to set system paths and variables and the creation of folders. Internet access appreciated.

Privileges at run time	The WISKI7 OS user must have write access to the configuration folder of WISKI7. During a WISKI7 session, local configuration files will be updated.
Installation folder	The local installation folder can be chosen individually.
Connection to the Meta Data Database (ORACLE)	<p>ORACLE Client 18 / 19 each with the latest patch, Net Service Name to WISKI7 Meta Data Database Server. The versions 10gR2 and 11g are no longer supported by ORACLE.</p> <p>Note: The WISKI7 client requires a 32-bit client, and the connection parameters for the database server must be set to character sets that match your specific language encoding, e.g. for German "Umlaute":</p> <p>NLS_CHARACTERSET = WE8ISO8859P15</p> <p>NLS_NCHAR_CHARACTERSET = UTF8</p> <p>For other language encodings different character sets are relevant, see the link Database Globalization Support Guide to check your language settings and contact your system administrator if errors occur with DbMaintain.exe.</p> <p>Note: The WISKI Client is a 32-bit application. The Extended Support for the Oracle 32-bit Client starts May 02, 2024, it is free of charge until April 30, 2026 and ends (then subject to a fee) on April 30, 2027.</p> <p>Database Releases and Support Timelines</p> <p>Source: https://blogs.oracle.com/ebstech/post/upgrade-support-for-oracle-database-12102-and-11204</p>
Connection to the Meta Data Database (MSSQL)	ODBC Microsoft SQL Server (version 17 or higher, 'SQL Server' driver no longer sufficient), DSN entry (system) to WISKI7 Meta Database
Connection to the Meta Data Database (PostgreSQL)	ODBC PostgreSQL, DSN Entry (System) to WISKI7 Meta Database
Connection to KISTERS Dispatcher	Port 7430
Connection to KISTERS WISKI server	Port is configurable at installation time. See the default port settings on server side.
Mandatory third party software	Adobe Acrobat Reader, GPL Ghostscript 8.70, 32-bit (optional for using German year book page (DGJ) reports, program name: gs870w32.exe).

Note:

- For web browser recommendations for the WISKI Standard Portal, see ch. [Using the WISKI Standard Portal](#)^[31].
- Keeping your web browser up to date is important for both security and correct display of web pages. Therefore it is mandatory to use a current or the latest available browser version to access Keycloak, TSM WebAdmin and other administrative web applications.

4.2.2 Installation Requirements

It is recommended to install your WISKI7 application under the KISTERS root directory `c:\kisters`.

Note the additional information about the corresponding INI files:

- [WISKI.ini](#)^[31]
- [WISKI_user_<user>.ini](#) and [WISKI_prof_<user>.ini](#)^[31]

4.2.2.1 WISKI.ini

By default the WISKI7 client expects the **WISKI.ini** file in the client application directory itself.

Existing customised **WISKI.ini** files can also be copied to the application directory.

If the application directory is protected from write processes or is located on a central machine, a **redirect.ini** might be created pointing to the external **WISKI.ini**.

```
[redirect]
app_ini_path=<External/Local path>
```

4.2.2.2 WISKI_user_<user>.ini and WISKI_prof_<user>.ini

By default the WISKI7 client expects the files **WISKI_user_<user>.ini** and **WISKI_prof_<user>.ini** in the client application directory itself.

Already existing customised files **WISKI_user_<user>.ini** and **WISKI_prof_<user>.ini** can be linked in the **WISKI.ini**.

```
[Paths]
UserIniPath=C:\temp\
```

4.3 Databases and Storage

The following databases are supported.

Hardware	The hardware requirements differ from installation to installation. Depending on import frequency and other factors a hardware recommendation could differ vastly.
ORACLE	<p>ORACLE 18 / 19; administrator privileges, WISKI7 Database as Schema-Owner; Ideally WISKI7 will use the default tablespace KISTERS.</p> <p>Note: ORACLE 12.2.0.2 or higher, also known as ORACLE 18.x/ 19.x is now also supported.</p> <p>For KiTSM/ WISKI7 only the <i>pure</i> database server is used. Deselect all options: JAVA, Multimedia ... (including their tablespaces). Just keep the necessary options for the Enterprise Manager. We recommend the following settings for a <i>typical</i> database server used for WISKI7:</p> <p>Note: Depending on your individual load profile (e.g. number and frequency of data import, complexity of time series structures, etc.) other configurations may be recommended.</p>

	<ul style="list-style-type: none"> ▪ DB_BLOCK_SIZE = 16384 (depending on the OS and how the HDDs are formatted, if an OS different than MS-Windows is being used, consider a DB_BLOCK_SIZE of 32k) ▪ PROCESSES ≥ 550 (depending on the system, if a larger amount of users and/ or processes are involved in such a system, these numbers (including SESSIONS) have to be increased) ▪ SESSIONS ≥ 500 (s. PROCESSES) ▪ REDOLOGSIZE per group from 500 MB to 4 GB ▪ SGA ≥ 4 GB on 64-bit OS ▪ PGA ≥ 1 GB on 64-bit OS <p>SGA/ PGA depend on the size of the database and the hardware configuration. For 11g or higher choose the global memory option (MEMORY_TARGET with 4 GB to 196 GB on 64-bit OS) without individual specifications of PGA/ SGA)</p> <ul style="list-style-type: none"> ▪ CHARACTER SET = WE8ISO8859P15 or similar 8-bit character set. <p>WISKI7 does not support UTF databases.</p> <p>Note: If the WISKI7 client is installed on a 64-bit operating system, you must additionally install the 32-bit database client to access the ORACLE TCA database.</p>
MSSQL	MSSQL Server 2016 / 2019 / 2022; administrator privileges, WISKI7 database as DB-Owner
PostgreSQL	PostgreSQL 15.1
Firebird (only standalone software versions like BIBER, KiWQM)	Firebird databases are included in the installation packages of the WISKI7 standalone versions like BIBER, SKED or KiWQM.
Network connection	The database should be connected to a fast network with low response times. Recommended is a Bandwidth LAN 1000 MBit/s and database response times of 10 milliseconds or less . Best results can be achieved when the WISKI server is installed on the same machine as the database server or is located in the same room as a direct network connection.

5 Preparations For a New Installation

The general installation steps are:

- Creating a database schema (which is the same as creating a user under ORACLE or a database under Microsoft SQL Server)
- Installation of KISTERS services including creation of necessary database tables
- Installation of the WISKI7 client
- Creating or updating the WISKI7 database

The installation steps are based on ZIP files provided on the KISTERS FTP server. See [Available packages](#)^[10] for details.



Note that a running database server (and a fair amount of expertise...) is a prerequisite for the steps described below.

The preliminary work comprises these steps:

- [Preparing the Database Server for the KISTERS WISKI Services](#)^[33]
- [Preparing the WISKI Server: Setup the security settings for the installation directory](#)^[44]
- [Preparing the WISKI Server: Install Perl](#)^[46]
- [Preparing the Client Machine: ORACLE Client Setup](#)^[49]

5.1 Preparing the Database Server for the KISTERS WISKI Services

It is important to understand at this stage that the WISKI7 database can be separated from the KiTSM database. However, this documentation assumes that both databases are installed in a common schema.

- Read the chapter [Installing Databases under ORACLE](#)^[33] to learn how to install the WISKI7/KiTSM database under ORACLE.
- Read the chapter [Installing Databases under Microsoft SQL Server](#)^[34] to learn how to install the WISKI7/KiTSM database under MSSQL.
- Read the chapter [Installing Databases under PostgreSQL](#)^[42] to learn how to install the WISKI7/KiTSM database under PostgreSQL.
- Read the chapter [WISKI7 Client Installation Tasks](#)^[104] to learn how to install WISKI7.

Note: If you have imported your database from an existing WISKI7/KiTSM database, refer to the topic about [Duplicating KISTERS WISKI server database](#)^[89].

Detailed information on how to install the WISKI7/KiTSM database under ORACLE or MSSQL can be found in the next subchapters:

- [Installing Databases under ORACLE](#)^[33]
- [Installing Databases under Microsoft SQL Server](#)^[34]
- [Installing Databases under PostgreSQL](#)^[42]

5.1.1 Installing Databases under ORACLE

Follow these instructions when installing ORACLE databases:

- [Create Tablespace](#)^[33]
- [Create User](#)^[34]

5.1.1.1 Create Tablespace

Copy and paste the following statement to an ASCII editor and store it under an appropriate name, e.g. **createTBS.sql**.

Replace the italic terms with the appropriate content (such as **D:\ORACLE\oradata\ORCL** instead of **\$DATAFILEPATH**).

```
SPPOOL result_createTBS.log
```

```
CREATE TABLESPACE WISKI LOGGING DATAFILE '$DATAFILEPATH\WISKI1.dbf'
SIZE 1000M REUSE AUTOEXTEND ON NEXT 200M MAXSIZE 10000M,
'$DATAFILEPATH\WISKI2.dbf' SIZE 1000M REUSE AUTOEXTEND ON NEXT 200M
MAXSIZE 10000M EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT
AUTO;
exit
```

5.1.1.2 Create User

Copy and paste the following statement to an ASCII editor and store it under an appropriate name, e.g. `createWISKIUSER.sql`. Replace the italic terms with the appropriate content (such as **WISKIADM** instead of **\$WISKIDBUSER**).

Note: Use only letters and numbers, e.g. **wiskiadm** is better than **w7-user**.

```
SPOOL result_createWISKIUSER.log
CREATE ROLE RESOURCE2 NOT IDENTIFIED;
GRANT CREATE CLUSTER TO RESOURCE2;
GRANT CREATE INDEXTYPE TO RESOURCE2;
GRANT CREATE MATERIALIZED VIEW TO RESOURCE2;
GRANT CREATE OPERATOR TO RESOURCE2;
GRANT CREATE PROCEDURE TO RESOURCE2;
GRANT CREATE ROLE TO RESOURCE2;
GRANT CREATE SEQUENCE TO RESOURCE2;
GRANT CREATE TABLE TO RESOURCE2;
GRANT CREATE TRIGGER TO RESOURCE2;
GRANT CREATE TYPE TO RESOURCE2;
GRANT CREATE VIEW TO RESOURCE2;
GRANT SELECT ON SYS.V_$SESSION TO RESOURCE2;
GRANT SELECT ON SYS.V_$SESSTAT TO RESOURCE2;
GRANT SELECT ON SYS.DBA_DATA_FILES TO RESOURCE2;
CREATE USER $WISKIDBUSER IDENTIFIED BY "12years" DEFAULT TABLESPACE
WISKI QUOTA UNLIMITED ON WISKI ACCOUNT UNLOCK;
GRANT CONNECT TO $WISKIDBUSER;
GRANT RESOURCE2 TO $WISKIDBUSER;
exit
```

Open command shell to Command Prompt (RUN/CMD.EXE) and change to the script directory.

```
>sqlplus -l sys/syspassword@instancealias as sysdba @createTBS.sql
>sqlplus -l sys/syspassword@instancealias as sysdba
@createWISKIUSR.sql
```

Check log files for errors.

5.1.2 Installing Databases under Microsoft SQL Server

Follow these instructions when installing databases on Microsoft SQL Server:

- [Creating a Database](#) 
- [Connecting to the Server](#) 
- [Create a new Database](#) 
- [Create a Login](#) 
- [Create New User](#) 
- [Maintenance and Monitoring](#) 

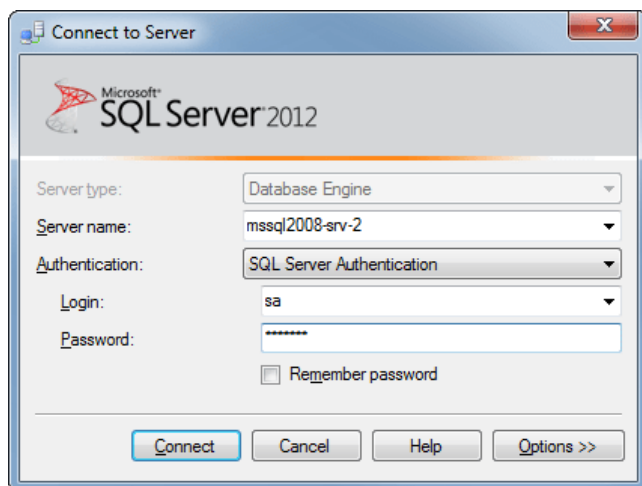
5.1.2.1 Creating a Database

Start the Microsoft SQL Server (**START > MICROSOFT SQL SERVER > SQL SERVER MANAGEMENT STUDIO**).

5.1.2.2 Connecting to the Server

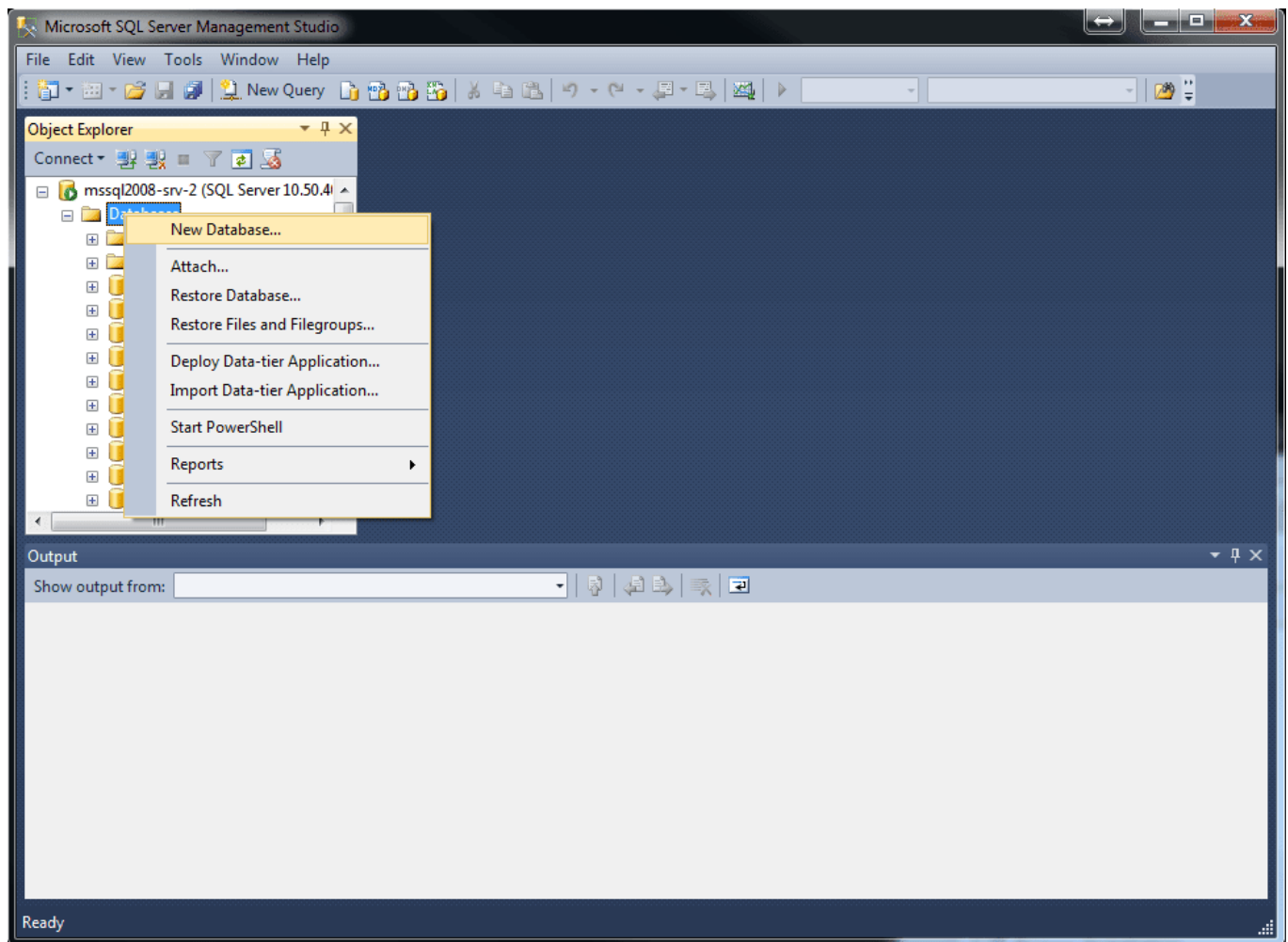
Server type	Database Engine	
Server Name	\$SQL_SERVER	<p>\$SQL_SERVER is the name of the SQL server which should host the WISKI7 database.</p> <p>In this example the SQL Server is called mssql2008-srv-2. This name will appear in all the screenshots.</p>
Authentication	SQL Server Authentication	
Login	sa	
Password	\$SA_PASSWORD	Log in as sa . If you have installed the MSSQL Server before, you know the content of the variable \$SA_PASSWORD .

However, you may need to ask the system administrator for your login data.



5.1.2.3 Create a new Database

To create a new database, you have to open one of the **SQL Server** displayed in the Object Explorer of the **SQL SERVER MANAGEMENT STUDIO**. Select the folder **DATABASES** and select the **NEW DATABASE** context menu item.



The database configuration dialog opens. Enter the database name (e.g. `$database_name=wiski`). Set the initial size to 128 MB. The database should be set to be automatically incremented by 16 MB (unlimited).

New Database

Select a page: General, Options, Filegroups

Script Help

Database name:

Owner:

☒ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
wiski	Rows ...	PRIMARY	128	By 1 MB, Unlimited
wiski_log	Log	Not Applicable	2	By 10 percent, Unlimited

Connection:

Server: mssql2008-srv-2

Connection: sa

[View connection properties](#)

Progress: Ready

Add Remove

OK Cancel

New Database

Select a page: General, Options, Filegroups

Script Help

Collation:

Recovery model:

Compatibility level:

Containment type:

Other options:

Automatic	
Auto Close	False
Auto Create Statistics	True
Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	False

Cursor	
Close Cursor on Commit Enabled	False
Default Cursor	GLOBAL

Miscellaneous	
Allow Snapshot Isolation	False
ANSI NULL Default	False
ANSI NULLS Enabled	False
ANSI Padding Enabled	False
ANSI Warnings Enabled	False
Arithmetic Abort Enabled	False
Concatenate Null Yields Null	False
Cross-database Ownership Chaining Enabled	False

Allow Snapshot Isolation

OK Cancel

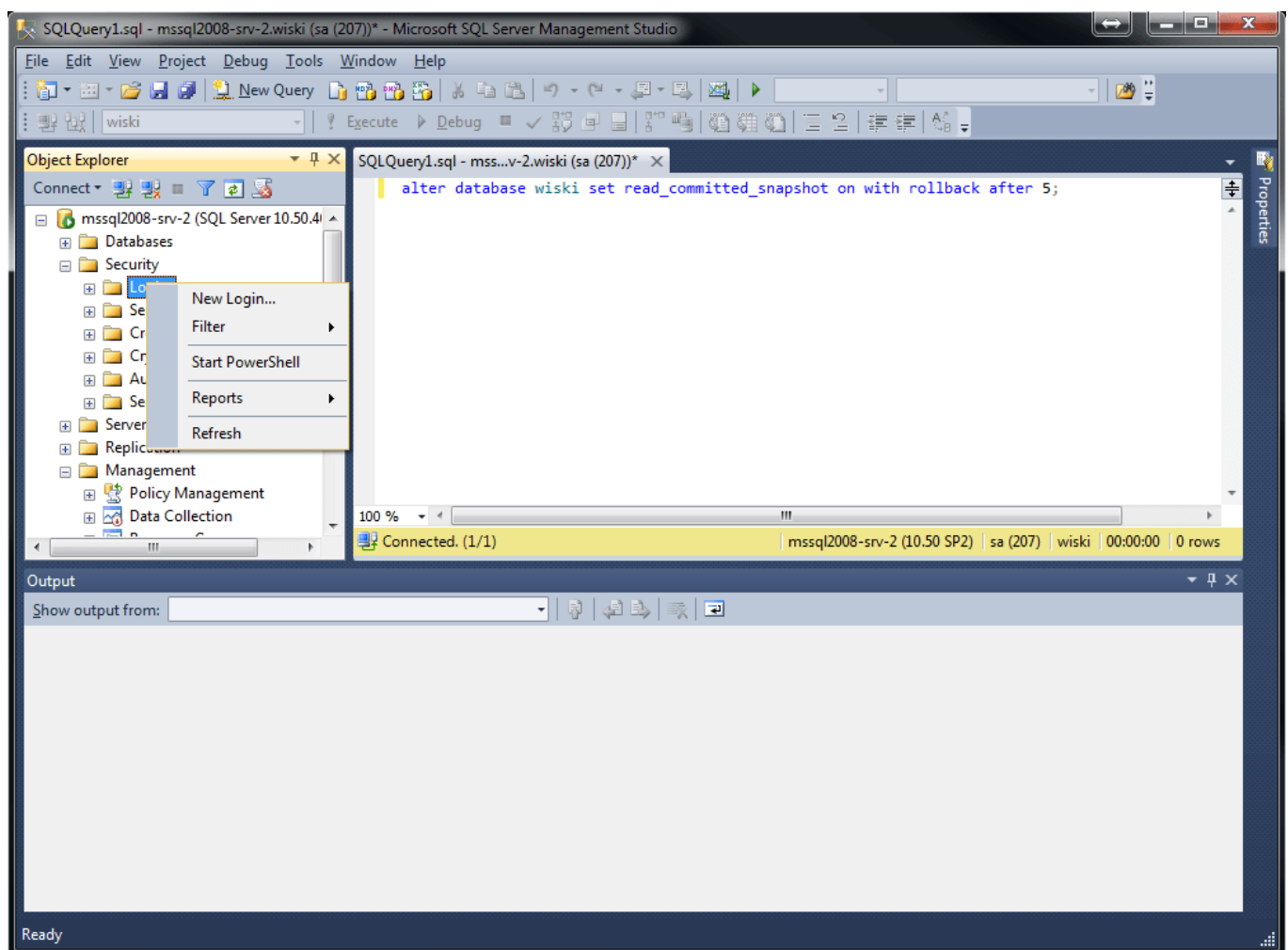
Note: Collation of the database must be configured correctly. A case-sensitive collation type must be selected, e.g. **GermanPhoneBook_BIN**, this can be done during the creation of the database (see screenshot) or afterwards via SQL `alter database $DATABASE_NAME collate German_PhoneBook_BIN`.

The next step is to set the database isolation level. KISTERS WISKI Services need a read-committed snapshot. To do this, activate the new query button and enter the following SQL to apply the setting. Replace `$DATABASE_NAME` with your database name.

```
alter database $DATABASE_NAME set read_committed_snapshot on with
rollback after 5;
```

5.1.2.4 Create a Login

Go back to the Object Explorer and create a new login. Select **SECURITY/LOGIN**, then select the **NEW LOGIN** context menu item. The login configuration dialog opens. Here you can configure the login name, password and default database (`$DATABASE_NAME` such as WISKI7).



Login name	\$W7LOGIN_NAME, e.g. wiskiadm
SQL Server authentication	
Password	\$W7LOGIN_PASSWORD
Confirm Password	\$W7LOGIN_PASSWORD
Enforce password expiration	uncheck this box

User must change password at next login	uncheck this box
Default database	\$DATABASE_NAME
Default language	ENGLISH

Login - New

Select a page: General, Server Roles, User Mapping, Securables, Status

Script Help

Login name: wiskiadm Search...

☐ Windows authentication

☒ SQL Server authentication

Password:

Confirm password:

☐ Specify old password

Old password:

☒ Enforce password policy

☐ Enforce password expiration

☐ User must change password at next login

☐ Mapped to certificate

☐ Mapped to asymmetric key

☐ Map to Credential

Add

Mapped Credentials

Credential	Provider
------------	----------

Remove

Default database: wiski

Default language: English

OK Cancel

Connection

Server: mssql2008-srv-2

Connection: sa

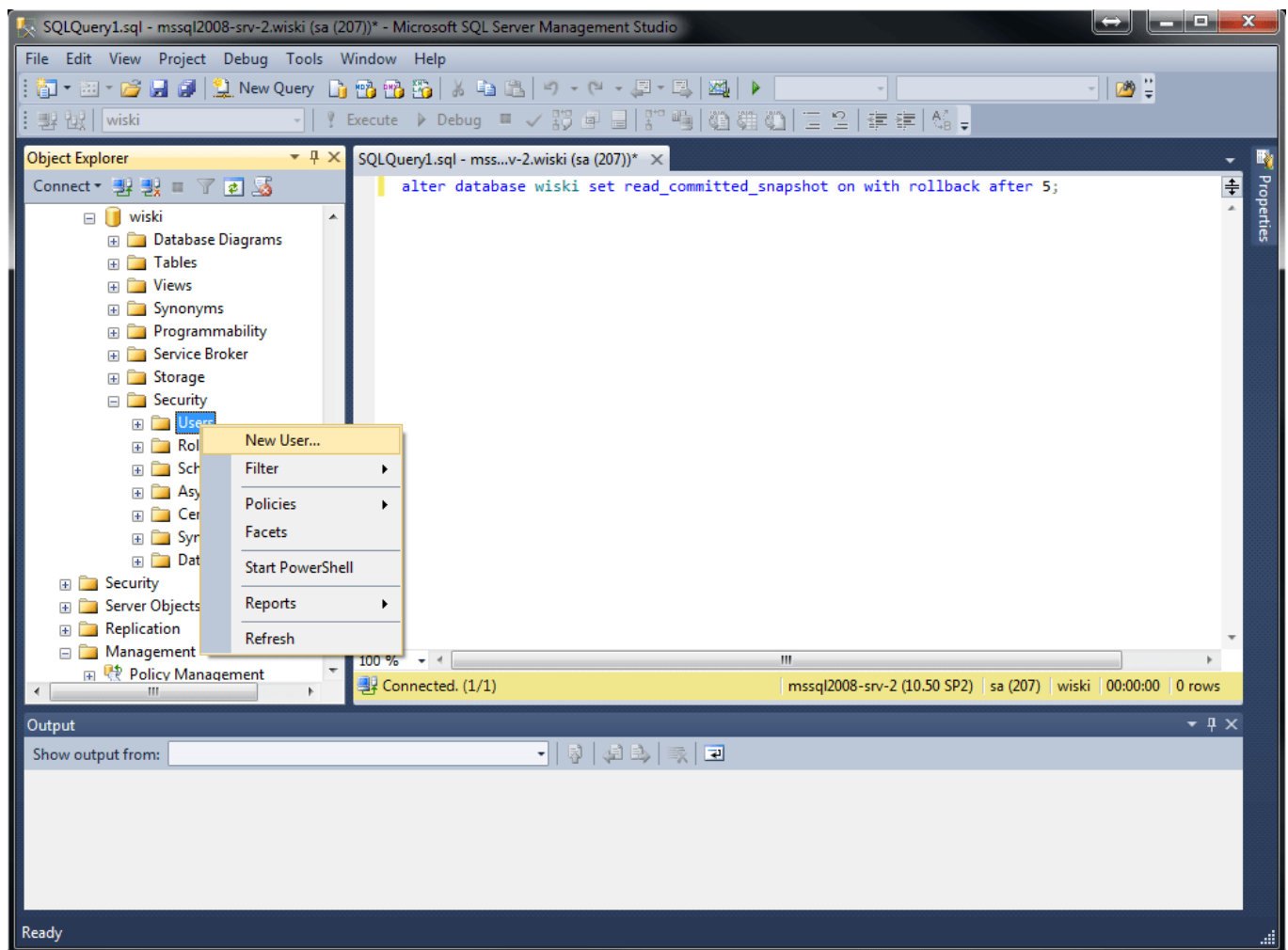
[View connection properties](#)

Progress

Ready

5.1.2.5 Create New User

Go back to the Object Explorer and create a new login. Select **DATABASES/\$DATABASE_NAME/SECURITY/USERS**, then select the **NEW USER** context menu item.



Database User - New

Select a page

- General
- Owned Schemas
- Membership
- Securables
- Extended Properties

Script Help

User type:
SQL user with login

User name:
wiskiadm

Login name:
wiskiadm

Default schema:
dbo

Connection

Server:
mssql2008-srv-2

Connection:
sa

[View connection properties](#)

Progress

Ready

OK Cancel

Database User - New

Select a page

- General
- Owned Schemas
- Membership
- Securables
- Extended Properties

Script Help

Schemas owned by this user:

Owned Schemas

- ☐ db_accessadmin
- ☐ db_backupoperator
- ☐ db_datareader
- ☐ db_datawriter
- ☐ db_ddladmin
- ☐ db_denydatareader
- ☐ db_denydatawriter
- ☒ db_owner
- ☐ db_securityadmin
- ☐ guest

Connection

Server:
mssql2008-srv-2

Connection:
sa

[View connection properties](#)

Progress

Ready

OK Cancel

5.1.2.6 Maintenance and Monitoring

On MSSQL servers it is necessary to configure some additional maintenance and monitoring:

1. Rebuild indexes and statistics on all tables in the schema on a regular basis. We recommend that you schedule a script to run every week (e.g. Saturday evening).
2. Monitor if there are long running transactions, these have a significant performance impact on the database.

```
select r.spid, r.login_time, r.last_batch, s.elapsed_time_seconds /
60 as "Minutes", t.text, r.hostname, r.program_name
FROM sys.sysprocesses r inner join
sys.dm_tran_active_snapshot_database_transactions s on r.spid =
s.session_id
CROSS APPLY sys.dm_exec_sql_text(r.sql_handle) t where
s.elapsed_time_seconds > 0
and program_name like 'KISTERS%'
```

5.1.3 Installing Databases under PostgreSQL

Follow these instructions when installing databases on PostgreSQL:

- [Install PostgreSQL server on Debian](#)⁴²
- [Install PostgreSQL server on WINDOWS](#)⁴³
- [Create User](#)⁴³
- [Important psql commands](#)⁴⁴

5.1.3.1 Install PostgreSQL server on Debian

Run the following commands to install PostgreSQL server on Debian.

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt
$(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc
| sudo apt-key add -
sudo apt -y update
sudo apt -y install postgresql-14 postgresql-client-14
sudo pg_createcluster 14 main
echo "host all all 0.0.0.0/0 md5" | sudo tee -
a /etc/postgresql/14/main/pg_hba.conf
echo "host all all ::0/0 md5" | sudo tee -
a /etc/postgresql/14/main/pg_hba.conf
sudo sed -ie
's/^max_connections/#max_connections/' /etc/postgresql/14/main/postg
resql.conf
echo "listen_addresses = '*' # listen on all IP addresses" | sudo
tee -a /etc/postgresql/14/main/postgresql.conf
echo "password_encryption = md5 # have to use md5 for WISKI, scram-
sha-256 is not working" | sudo tee -
a /etc/postgresql/14/main/postgresql.conf
echo "max_connections = 500 # required by WISKI" | sudo tee -
a /etc/postgresql/14/main/postgresql.conf
sudo pg_ctlcluster 14 main restart
sudo -u postgres psql -c "SELECT version();"

```

5.1.3.2 Install PostgreSQL server on WINDOWS

Preparation of the PostgreSQL database server**Note:** Make sure to have the latest PostgreSQL driver installed.

WISKI7 has problems connecting to a PostgreSQL server using the newer SCRAM-SHA-256 encryption. Therefore, we need to configure the PostgreSQL server to use MD5 encryption.

Note: If database users already exist and you change the encryption, you will need to set a new password for them.

Go to the PostgreSQL data folder, **C:\Program Files\PostgreSQL\14\data** by default on Windows.

First edit **postgresql.conf** as root user:

```
...
listen_addresses = '*'          # listen on all IP addresses
...
max_connections = 500          # required by WISKI
...
password_encryption = md5      # have to use md5 for WISKI, scram-
sha-256 is not working
...
```

Then edit **pg_hba.conf** as root user:

```
...
#allows all ipv4 addresses, change to subnet or whatever suites you.
host all all 0.0.0.0/0 md5
...
#allows all ipv6 addresses, change to subnet or whatever suites you.
host all all ::0/0 md5
...
```

Note: After these configuration changes you have to restart the server, a reload is not enough.

Warning: If you want to connect to the database from remote as database superuser (normally **postgres**), you have to assign a password to **postgres** first.

On database machine start **psql** as user **postgres** and type the following:

```
postgres=# \password postgres
```

You will be prompted to enter your password.

5.1.3.3 Create User

Optional step - Create tablespace

Not necessary, normally we use the default tablespace.

Could be done as database superuser (usually **postgres**) with the following command:

```
postgres=# CREATE TABLESPACE wiski_tbs LOCATION
'/ssd1/postgresql/wiski/data';
```

Create User

On the database server, login as database superuser (normally **postgres**) and start a **psql** session:

```

$ sudo -i -u postgres
$ psql
postgres=# \du

postgres=# create user wiskiadm password '*****' CREATEDB;

postgres=# select * from pg_catalog.pg_user;

postgres=# \l

postgres=# create database wiski OWNER wiskiadm LOCALE 'en_US.UTF-
8';
# if this leads to an error then use
#postgres=# create database wiski OWNER wiskiadm LOCALE 'en_US.UTF-
8' template 'template0';

postgres=# \q

```

5.1.3.4 Important psql commands

Below you can find some useful commands for PostgreSQL regarding WISKI.

```

# list tablespaces
postgres=# \db

# list databases
postgres=# \l

# list schemas
postgres=# \dn

#list roles
postgres=# \dg
postgres=# \du

# quit
postgres=# \q

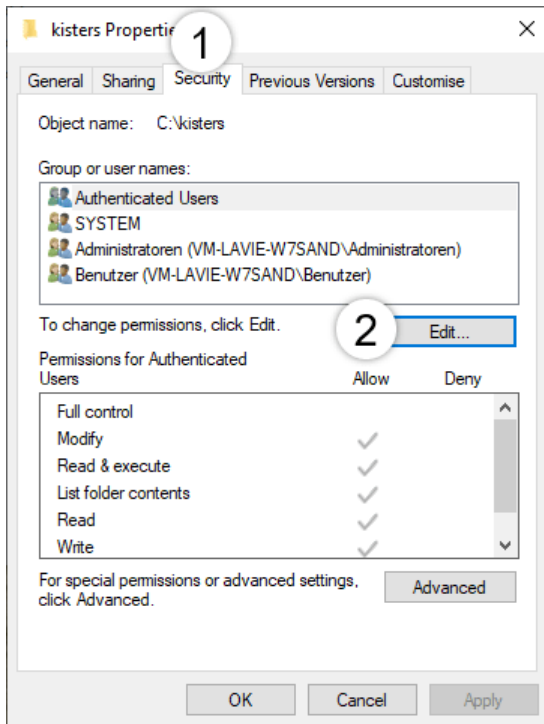
```

5.2 Preparing the WISKI Server: Setup the security settings for the installation directory

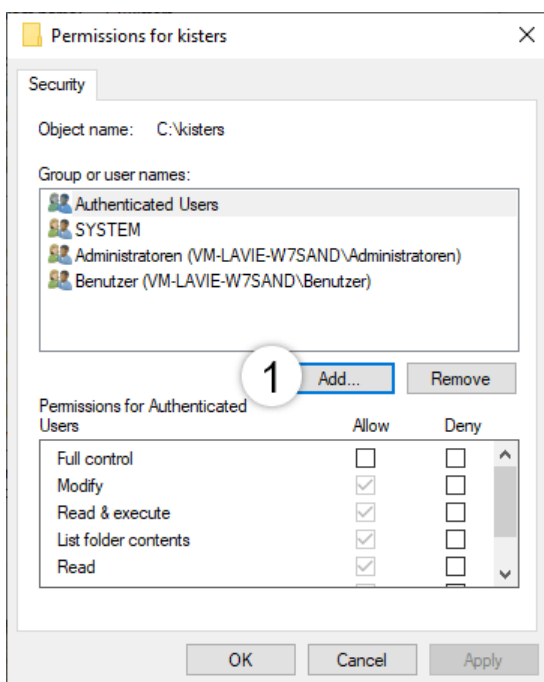
To set up the WISKI7 server, a root directory is needed for the installation. Usually the directory `c:\kisters\` is selected. This directory must be created manually and the WINDOWS user rights have to be set accordingly.

To set the user rights, proceed as follows:

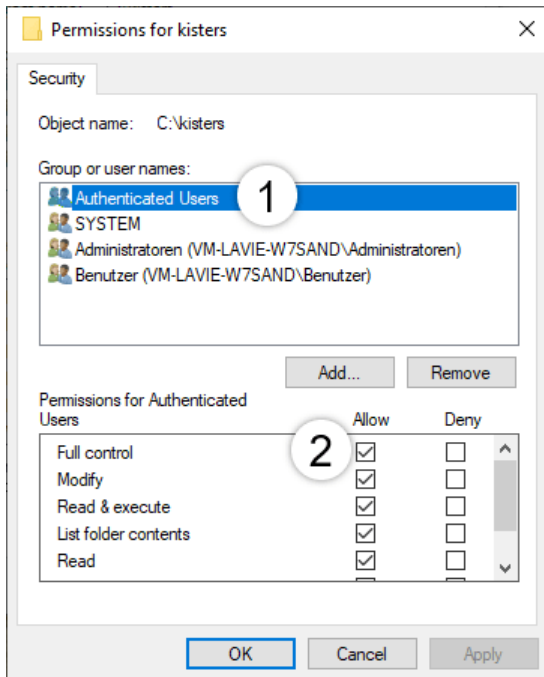
- Select the **PROPERTIES** context menu item of the directory.
- Select the **SECURITY** tab
- Edit the groups or users using the **[EDIT]** button (see screenshot below).



- Set the user rights to full control on the directories and files for all authenticated users.
- Use the **[ADD]** button to add the group of authenticated users (see screenshot below).



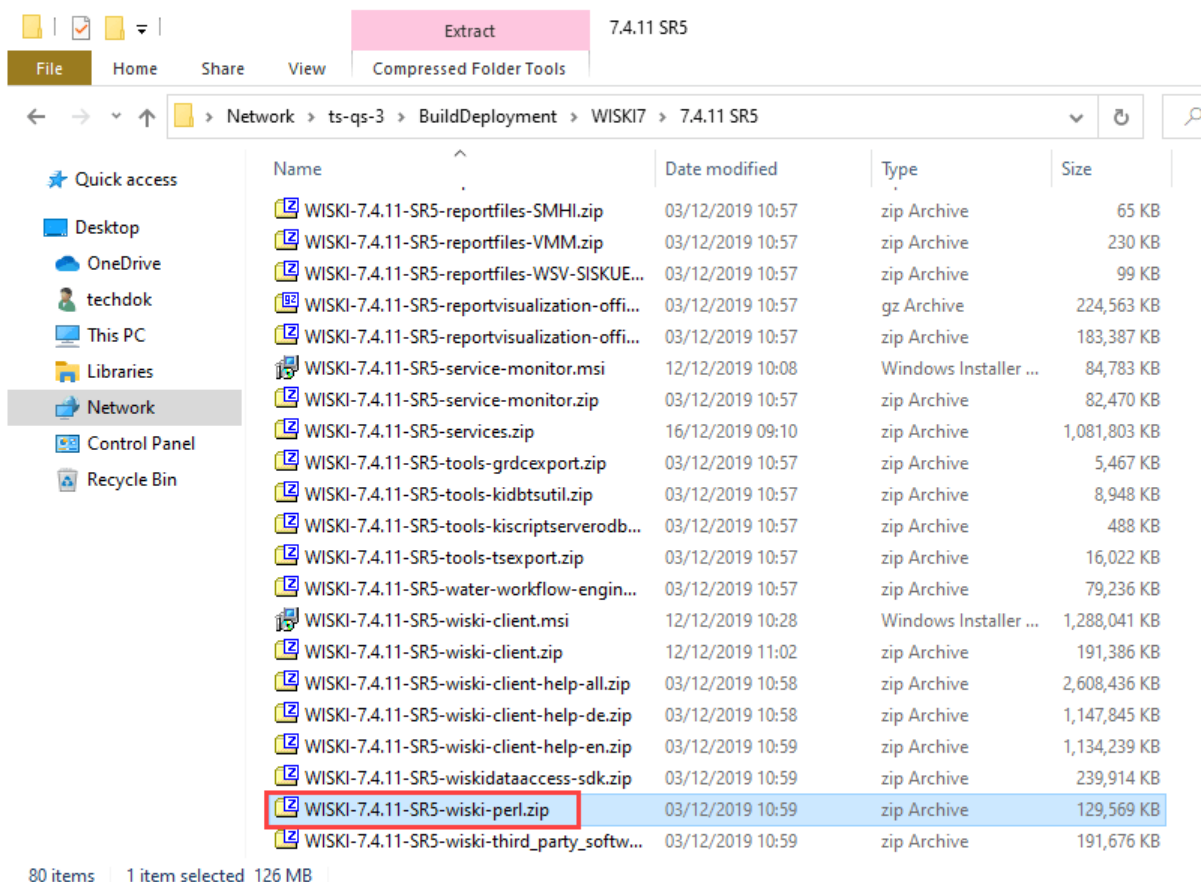
- Select the group and check the **FULL CONTROL** box (see screenshot below).



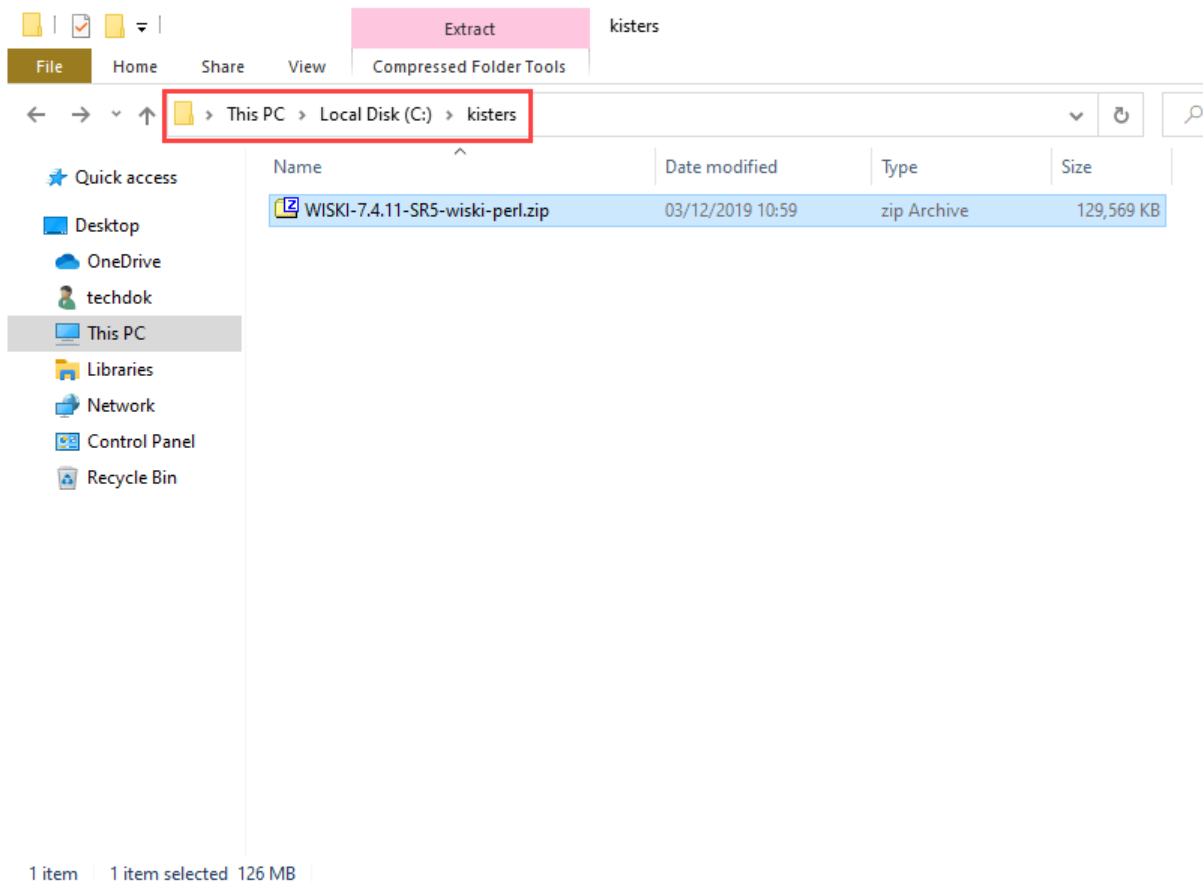
5.3 Preparing the WISKI Server: Install Perl

The KISTERS WISKI services require Perl to be installed. See the [Hardware and Software Requirements](#)^[13] for the KISTERS WISKI services.

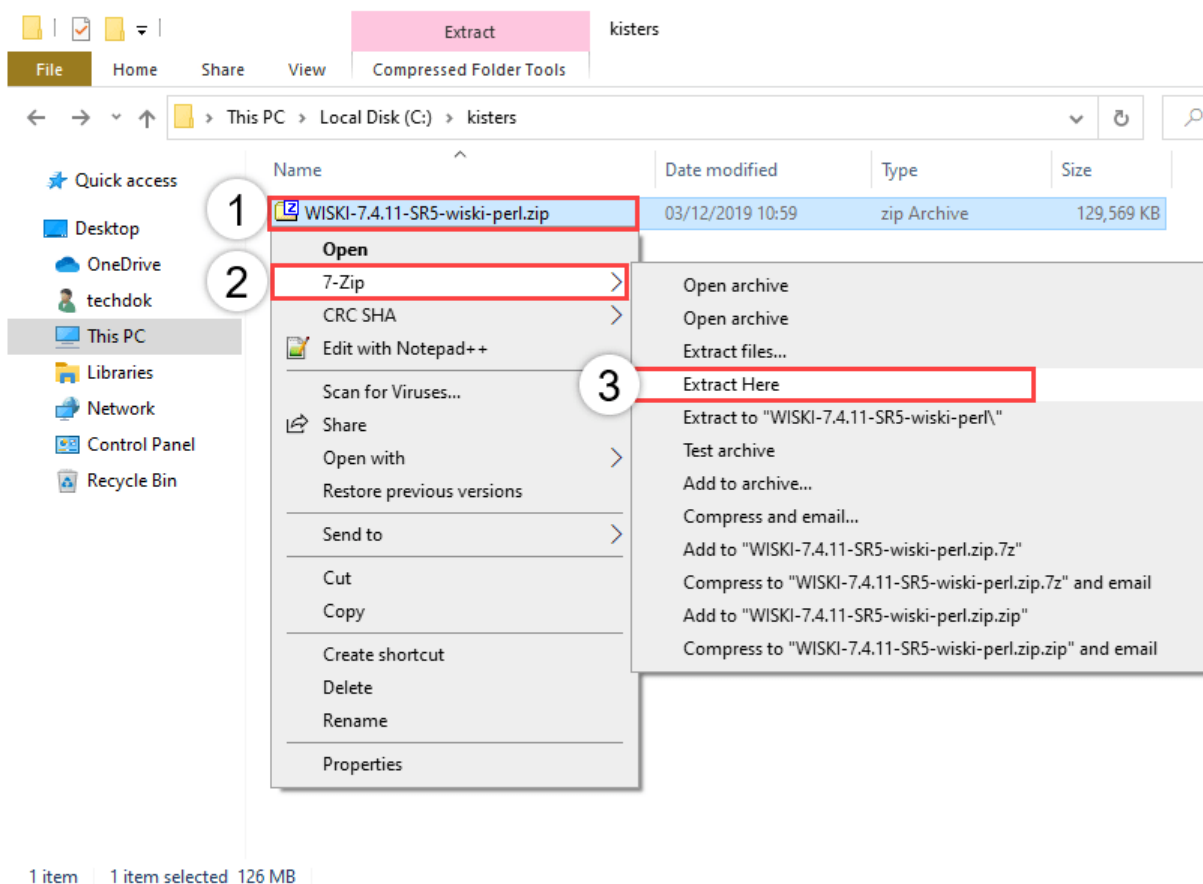
- Locate the Perl ZIP file **WISKI-<Version>-wiski-perl.zip** in the WISKI folder (see screenshot below).



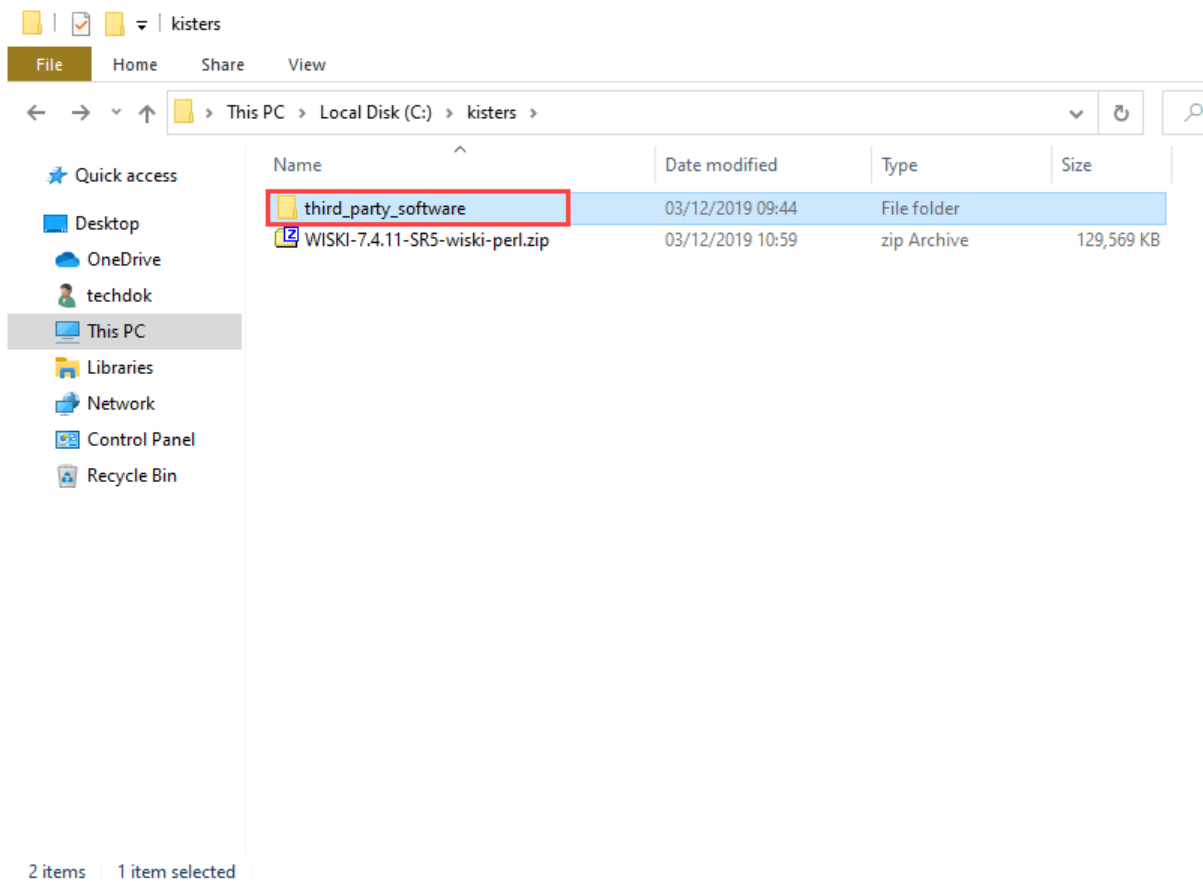
- Copy the ZIP file to your WISKI7 root directory, e.g. **c:\kisters** (see screenshot below).



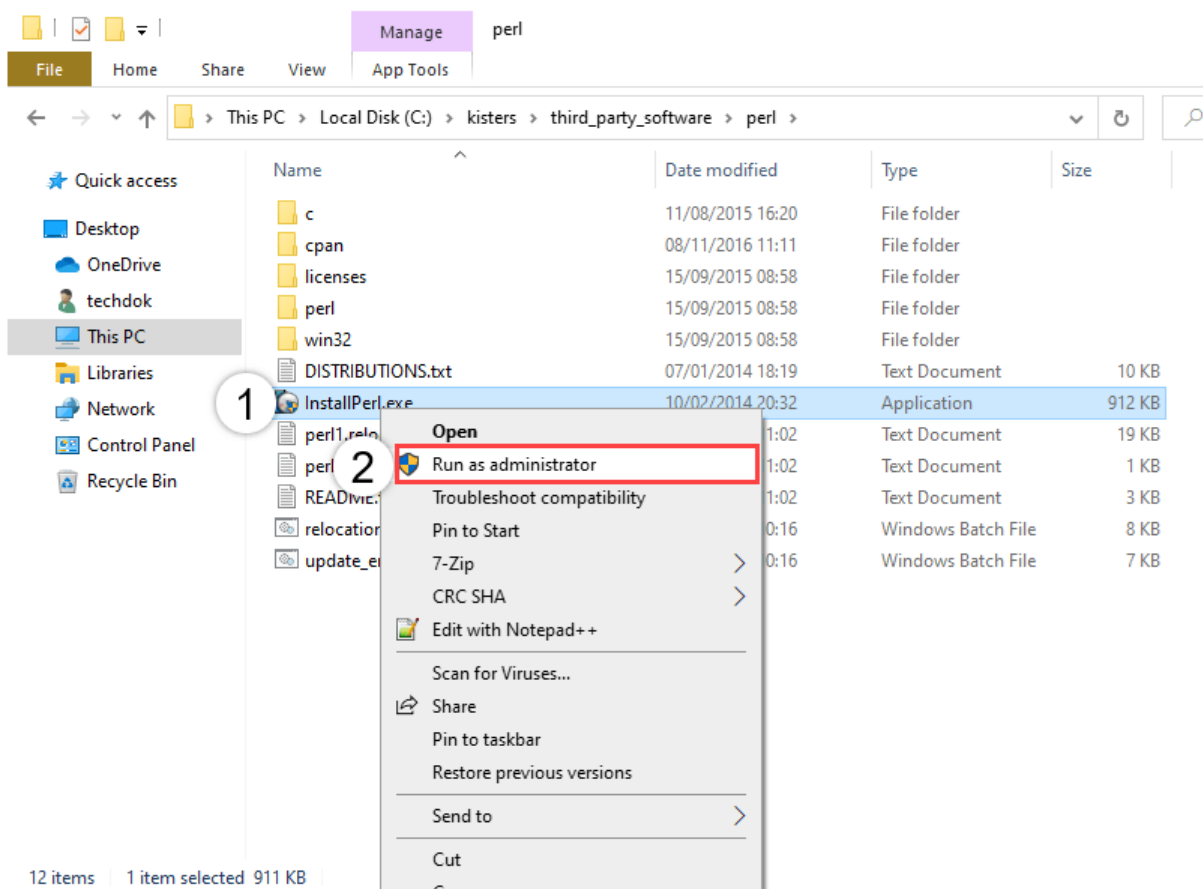
- Extract the ZIP file in this root directory (see screenshot below).



- Navigate to the `c:\kisters\third_party_software\perl` sub folder (see screenshot below).



- Execute **InstallPerl.exe** with administrator permissions (see screenshot below).

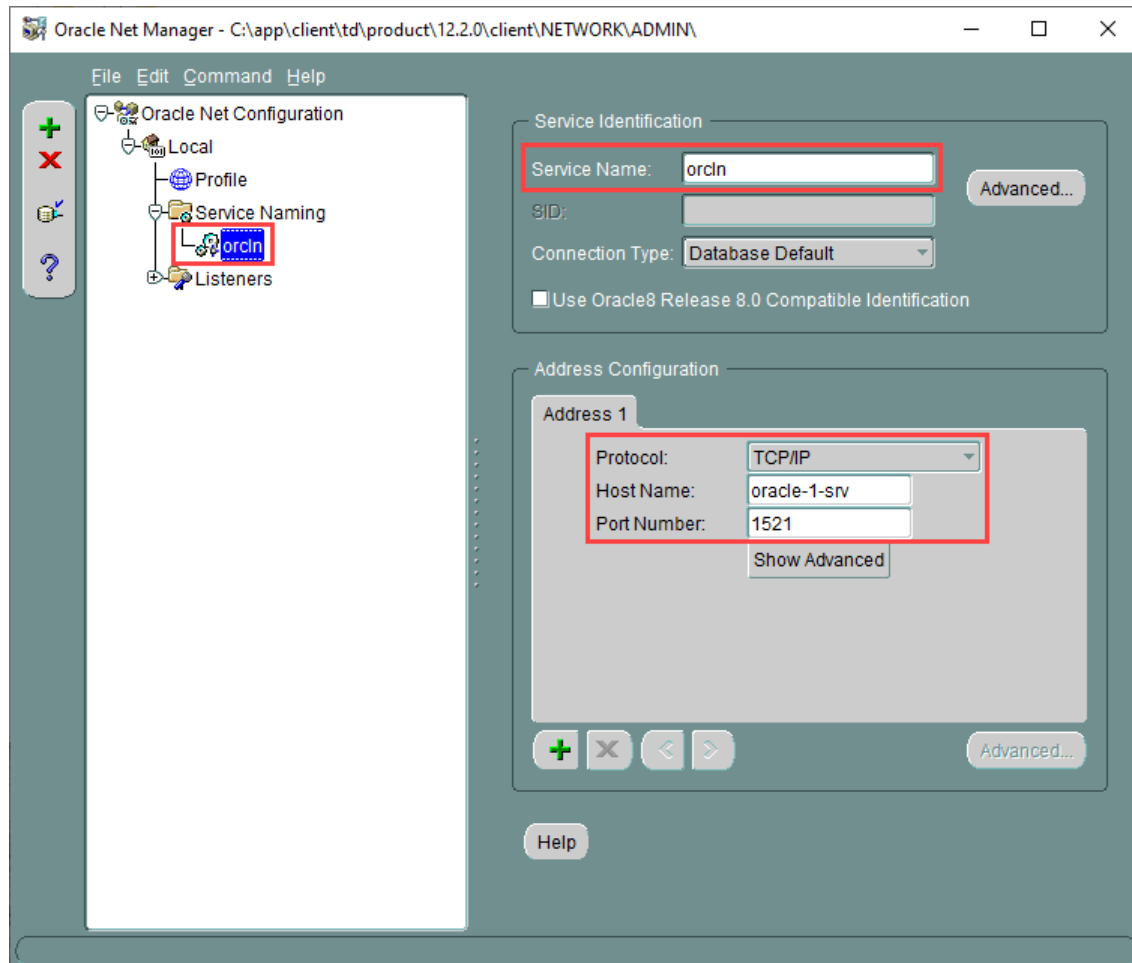


5.4 Preparing the Client Machine: ORACLE Client Setup

An ORACLE Installation of WISKI7 requires the ORACLE Client to be installed and configured.

Note: If the WISKI7 Client is installed on a 64-bit operating system, you must additionally install the 32-bit database client to access the ORACLE TCA database.

The client must be configured to point to the WISKI7 database server. ORACLE provides configuration tools for the Service Naming. Use the NetManager, for example. An example configuration under ORACLE for a server can be found below.



Note: Do not forget to save the network configuration (**FILE > SAVE NETWORK CONFIGURATION**).

6 WISKI Server Manager (WSM)

The WISKI Server Manager (WSM) is designed to perform a complete [initial WISKI installation](#)^[61] including the KISTERS WISKI server modules (KiTSM, KiIOSys, KiDSM, etc.) or to perform an [update installation](#)^[76]. The initial installation routine will ask for the database connection information. The different schemas will be deployed into one database. If the system is to be set up as a multi-tenant system or meet other requirements, the installation needs to be [done manually](#)^[100].

Detailed information can be found in the next subchapters:

[Deployed WSM Packages](#)^[50]

[WSM User Interface](#)^[51]

[Customise Actions](#)^[56]

[Log Viewer](#)^[60]

6.1 Deployed WSM Packages

The WSM consists of two packages:

- `WISKIServerManager.exe`
- `WISKIServerManager.zip`

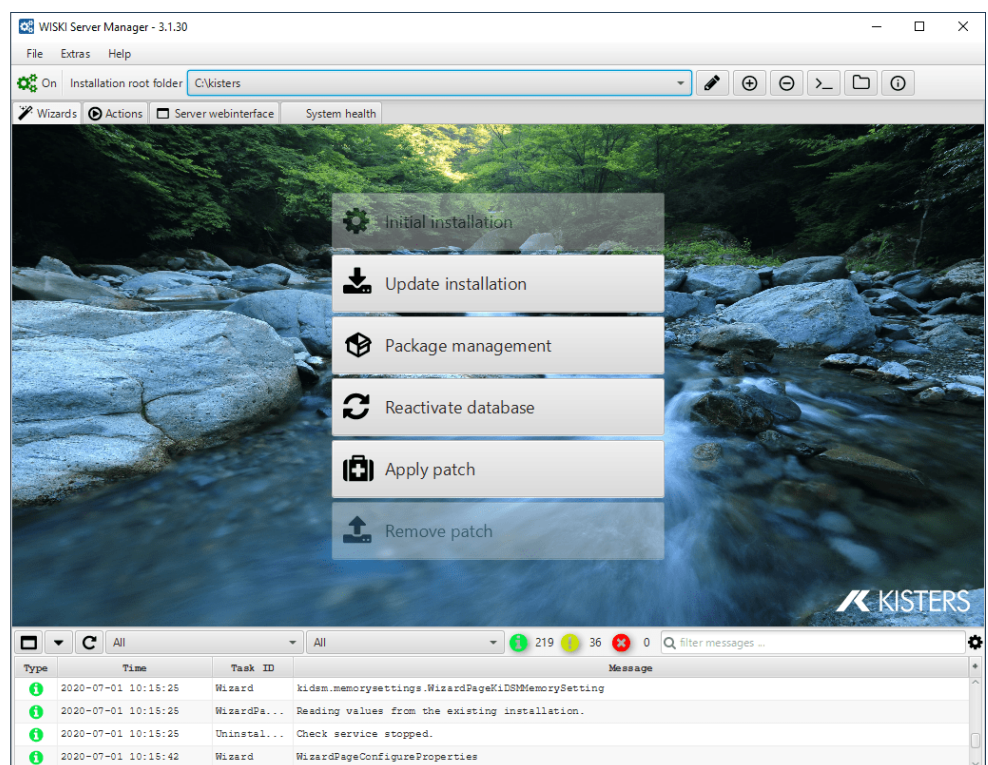
Save both packages in your installation root directory, e.g. `c:\kisters` and execute the `WISKIServerManager.exe`.

Note: The first start could take some time, because some files are being prepared. If the start fails with the error message **MAIN CLASS NOT FOUND**, use the batch file `<root>\WISKIServerManager\WISKIServermanager.bat` to launch the program.

Currently selected
software location

Available wizards

Logging area



- In the upper part of the main screen of the WISKI Server Manager, there is a symbol bar showing the directory of the current WISKI installation. The gear symbol in front of the software location indicates if the installed service is running.
- In the middle pane, you can see the available wizards that the WISKI Server Manager provides, for example, to initially install or update your system or to install certain components at a later date.
- At the bottom of the WISKI Server Manager is the logging area. This provides information about current events that are being logged in the WISKI Server Manager.

6.2 WSM User Interface

Mainly, the WSM user interface consists of menu items, several tabs (partly hidden by default) containing wizards, actions, a link to the TSM web interface and some system health metrics. The UI also includes a logging area, containing the [Log Viewer](#)⁶⁰.

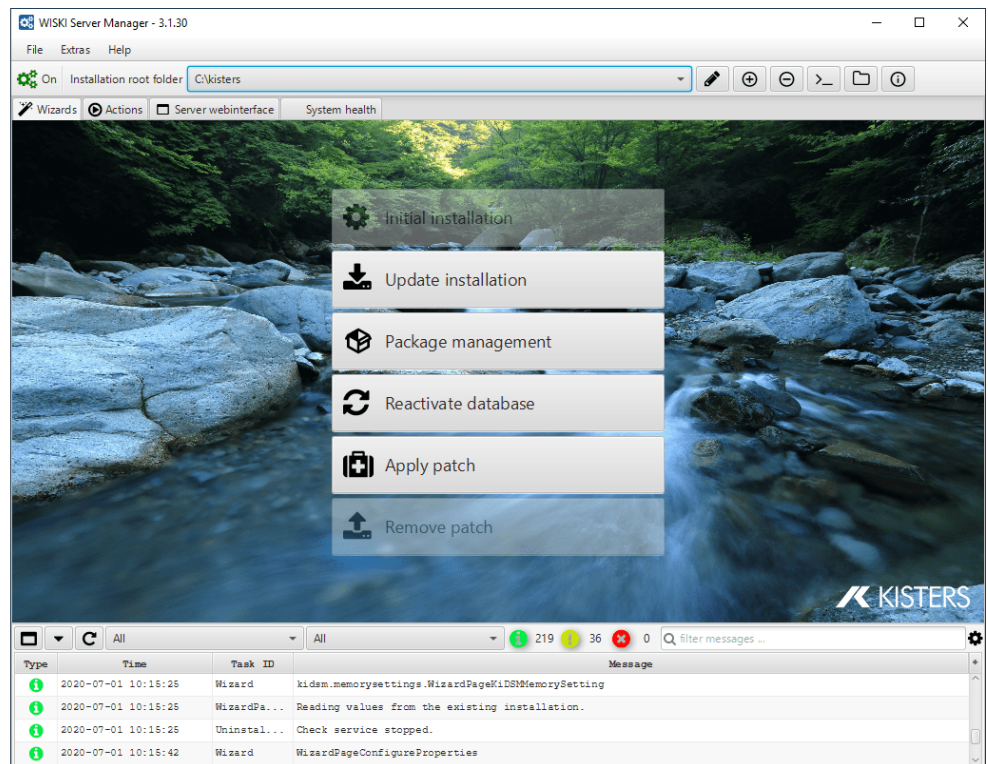
Currently selected software location



Available wizards



Logging area



We explain the WSM user interface in the next sup topics:

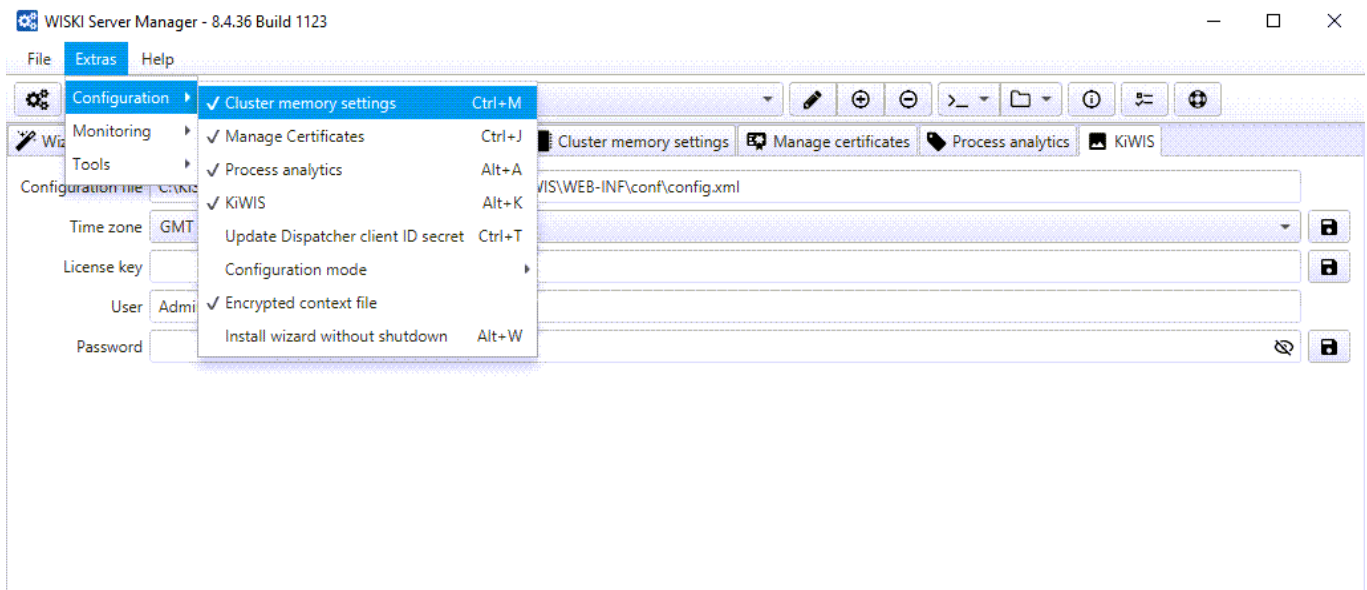
- [WSM Menus](#)⁵¹
- [WSM Tabs](#)⁵²

6.2.1 WSM Menus

under construction.

6.2.2 WSM Tabs

The functions of the WSM user interface are located, besides the menu items, in several tabs (partly hidden by default) containing wizards, actions, a link to the TSM web interface and some system health metrics.



Use the menu item **EXTRAS > CONFIGURATION** and the functions below to display or hide several tabs that are explained below (partly under construction).

Wizards

Actions

Server web interface

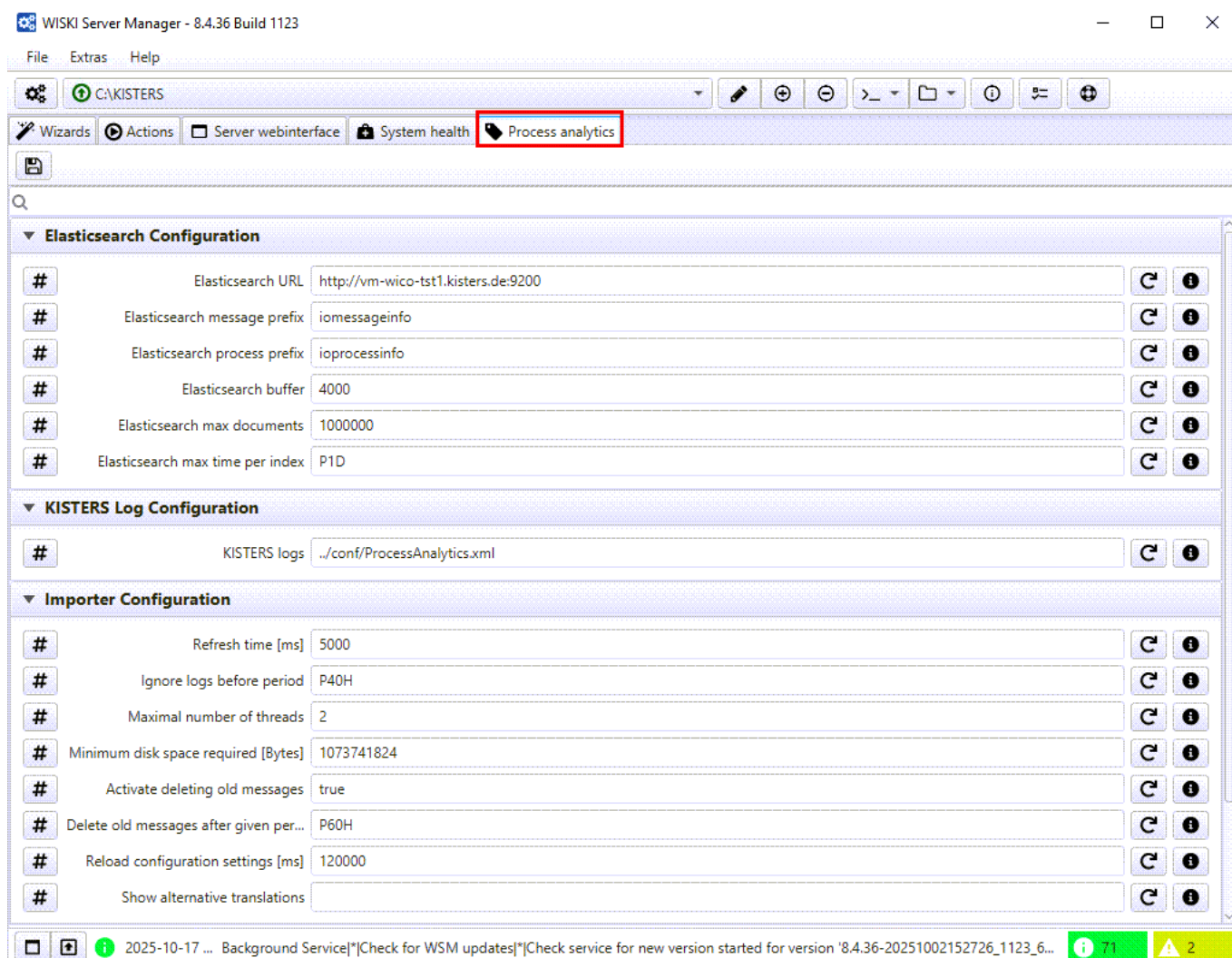
System health

Cluster memory settings

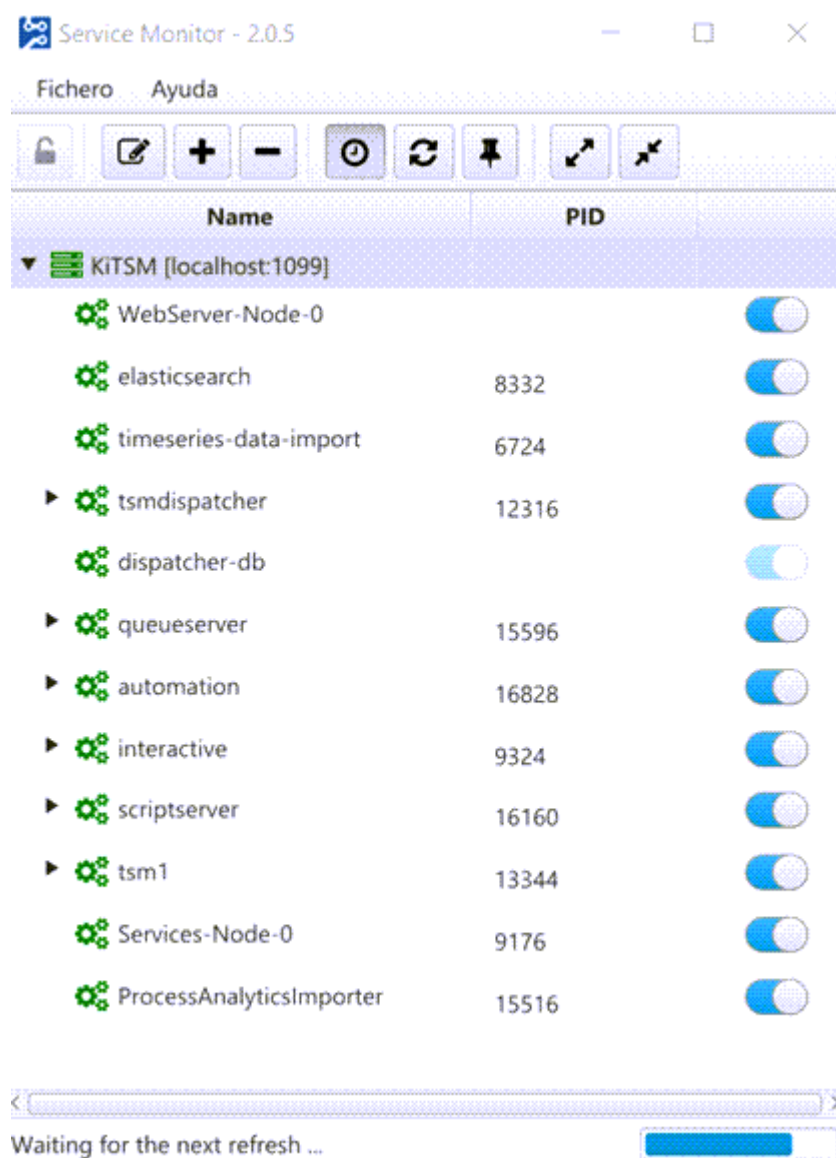
Certificates

Process Analytics

The **PROCESS ANALYTICS** tab is a supporting UI for the PCA configuration file `\kisters\wiski\services\processanalytics\conf\processanalytics-custom.yml`.



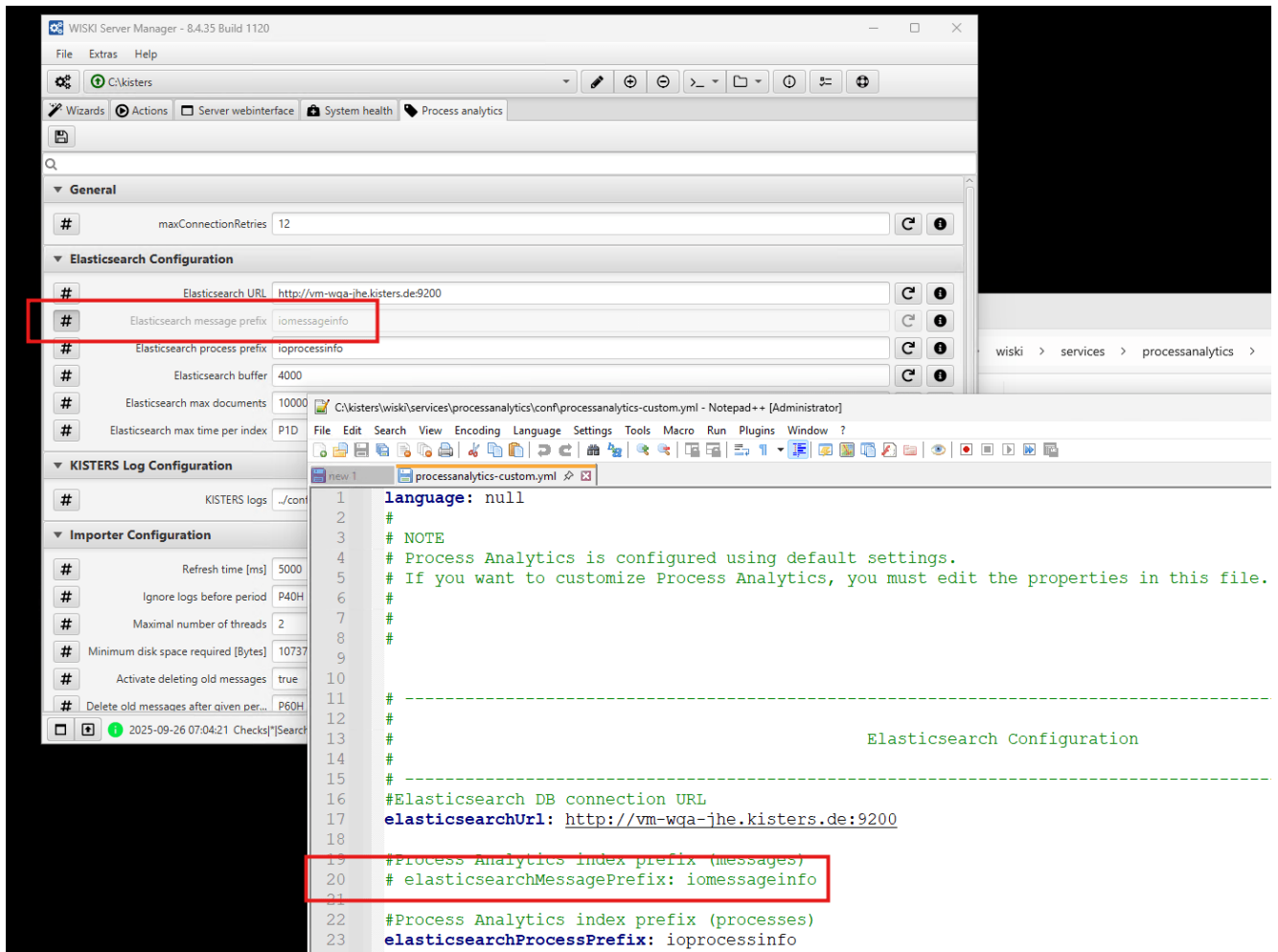
This configuration file is used by the PCA Importer (see the respective node in the [Service Monitor](#) ⁹¹).



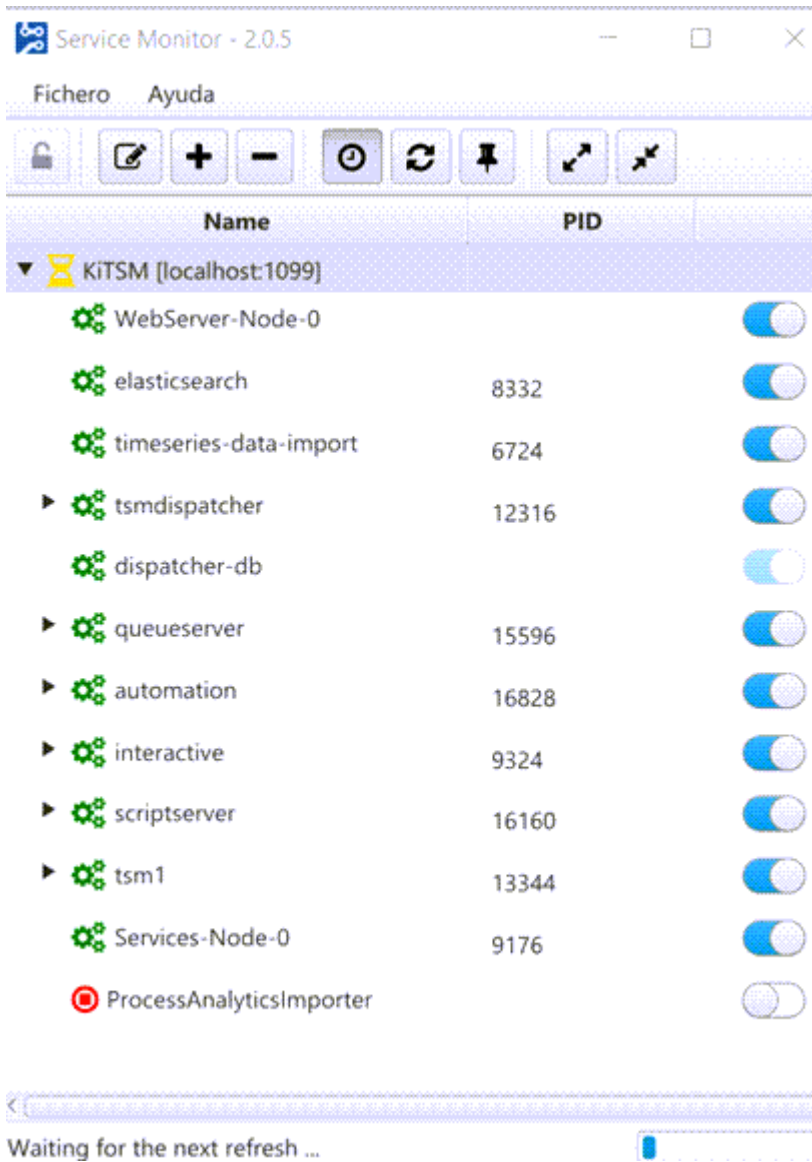
This service reads the configured log files on the fly and imports the log lines into the **elasticsearch** index.

Proceed like this:

- If you modify the Elasticsearch configuration yourself, you can click the hash button to comment out single parameters of the **processanalytics-custom.yml** file mentioned above:



- When you save changed settings, make sure that the ProcessAnalyticsImporter node in the Service Monitor is restarted so that the new configuration is loaded on startup.



KiWIS

Queue monitor

Session monitor

Update history

TSDIO monitoring

6.3 Customise Actions

Actions are stored in xml files. Each action has a name and a description, which are displayed on the **ACTIONS** tab.

Additionally each action has two JavaScript parts:

- Display condition
This part evaluates, if the action is to be shown on the main tab. This script must return true or false.
- Action script
This script defines what happens when the action is executed.

The WISKI Server Manager has two kinds of actions:

- Build-in actions (the configuration is saved in the installation directory)

- Customer actions (the configuration is saved in the %programdata% directory)

The build-in actions are updated together with the software. Customer actions stay untouched during an update.

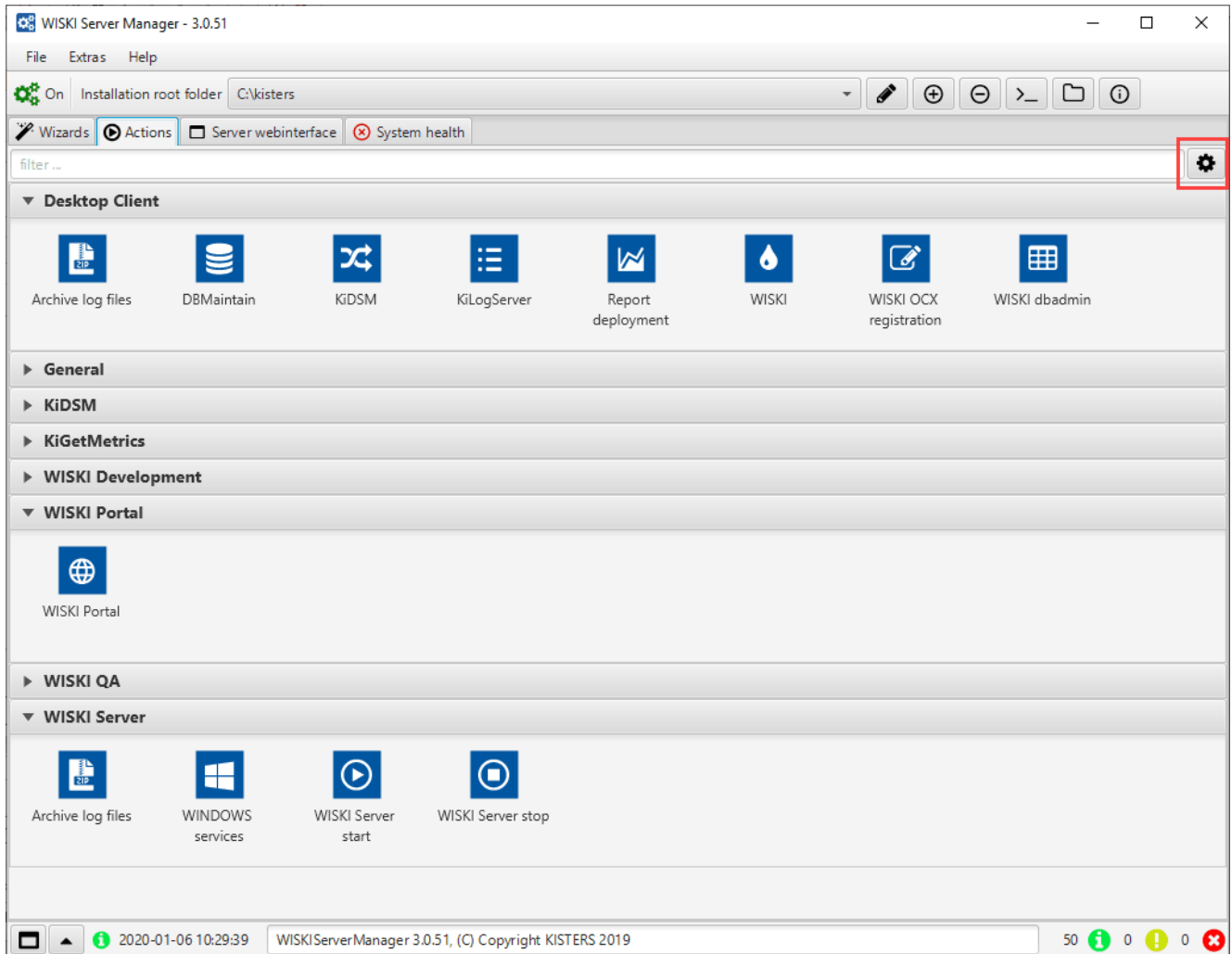
Built-in actions can be edited by clicking the unlock icon in the toolbar.

Note: Edited build-in actions will be overwritten during the next update.

Built-in actions can be copied as customer action, by clicking the respective icon in the toolbar.

The **ACTIONS** tab contains various actions to maintain the installation or to launch installed clients.

- Click the gear icon in the upper right corner to customise these actions (see screenshot below).

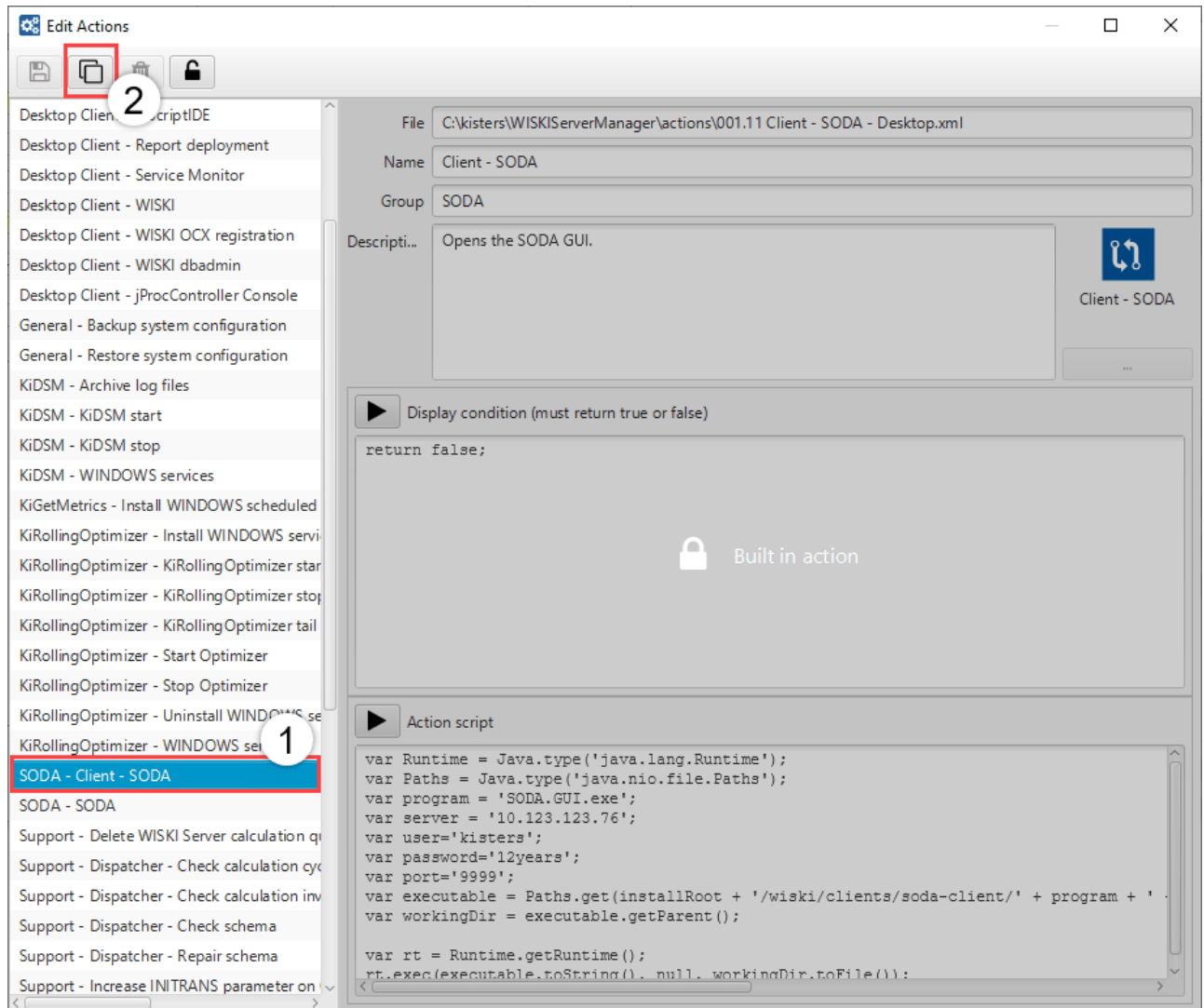


The configuration window opens. It shows all actions.

Example: Integrate SODA GUI

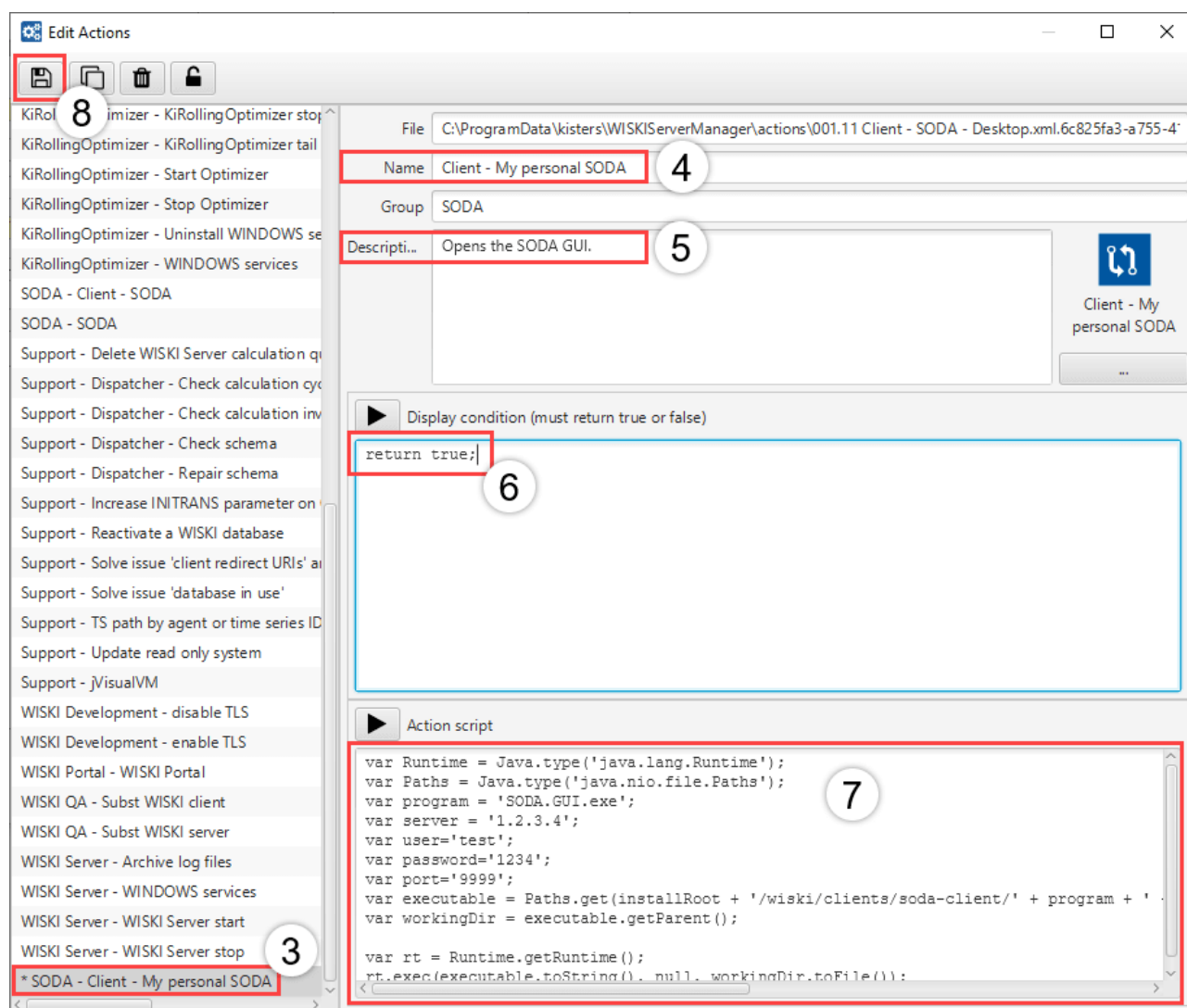
For the SODA GUI a script template already exists. To activate the action, the following steps have to be performed (see screenshot below).

- Select the template (1).
- Click the copy icon (2).



Now you have a custom action which can be modified as follows (see screenshot below).

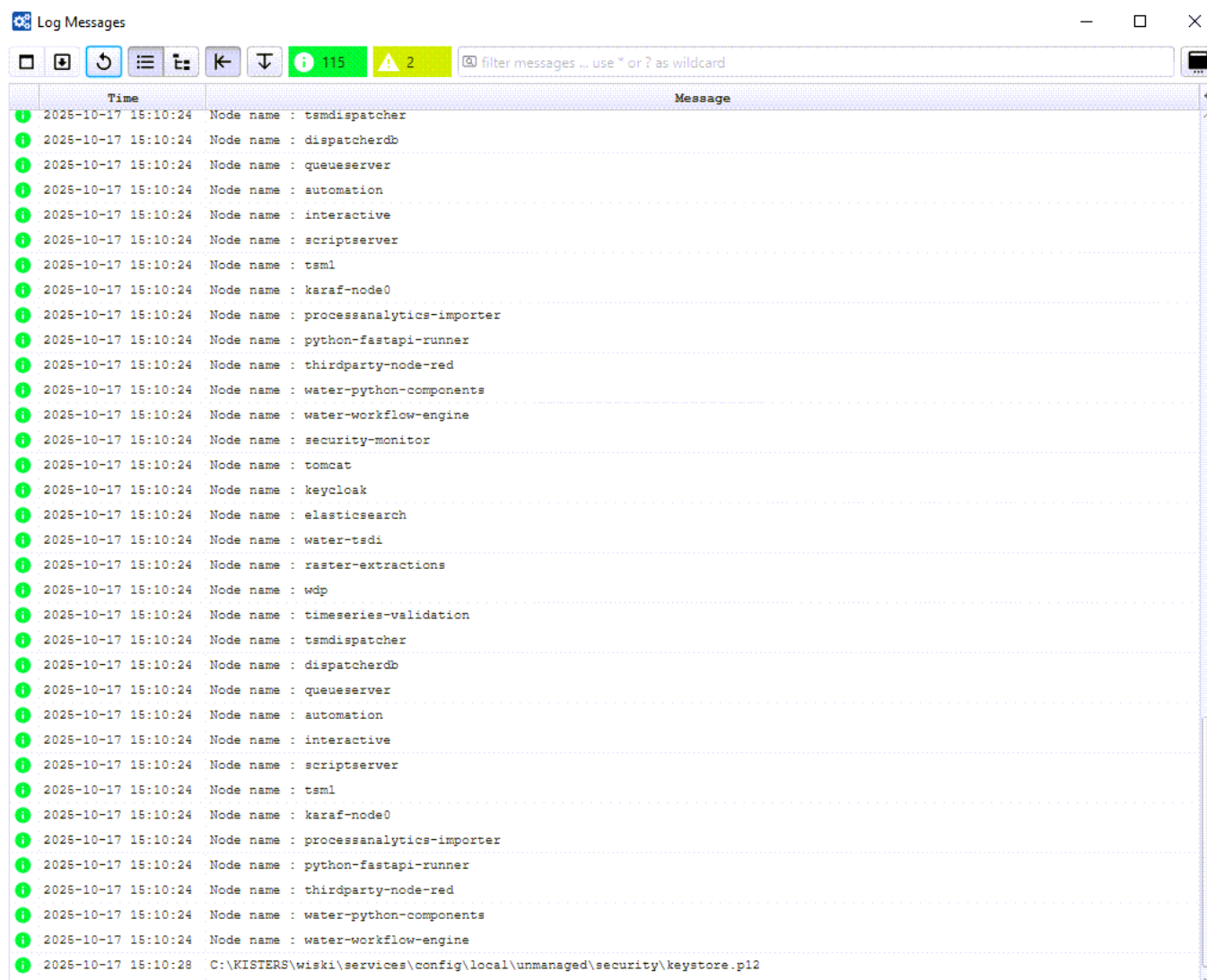
- Select the custom action (3).
- Adapt the name (4).
- Adapt the description (5).
- Change the return value of the display condition to true (6). This means the actions will always be shown.
Note: You can use the play icon to test the result.
- Change the **sodaIP** variable to the IP address of your SODA (7).
Note: You can use the play icon to test the result.
- Save the action and close the window (8).



The action list is now refreshed and sorted by name. By double-clicking the new action, the SODA GUI opens. If you have more SODAs, repeat the steps above.

6.4 Log Viewer

The LogViewer contains information on the WSM actions and behaviour.



The screenshot shows the 'Log Messages' window in WSKI Server Manager. The window has a toolbar with icons for filtering, refreshing, and navigating. A search bar contains the text 'filter messages ... use * or ? as wildcard'. The log messages are displayed in a table with two columns: 'Time' and 'Message'.

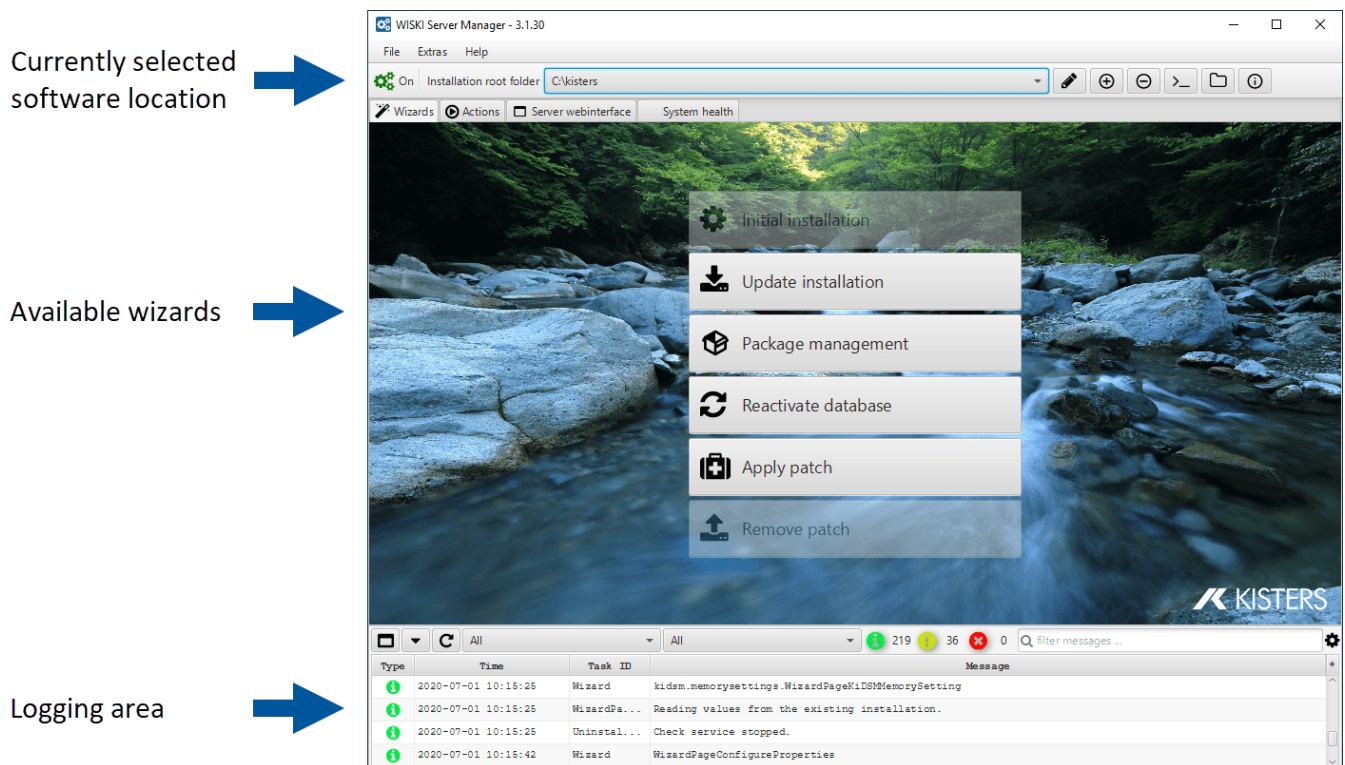
Time	Message
2025-10-17 15:10:24	Node name : tsmdispatcher
2025-10-17 15:10:24	Node name : dispatcherdb
2025-10-17 15:10:24	Node name : queueserver
2025-10-17 15:10:24	Node name : automation
2025-10-17 15:10:24	Node name : interactive
2025-10-17 15:10:24	Node name : scriptserver
2025-10-17 15:10:24	Node name : tsml
2025-10-17 15:10:24	Node name : karaf-node0
2025-10-17 15:10:24	Node name : processanalytics-importer
2025-10-17 15:10:24	Node name : python-fastapi-runner
2025-10-17 15:10:24	Node name : thirdparty-node-red
2025-10-17 15:10:24	Node name : water-python-components
2025-10-17 15:10:24	Node name : water-workflow-engine
2025-10-17 15:10:24	Node name : security-monitor
2025-10-17 15:10:24	Node name : tomcat
2025-10-17 15:10:24	Node name : keycloak
2025-10-17 15:10:24	Node name : elasticsearch
2025-10-17 15:10:24	Node name : water-tsdi
2025-10-17 15:10:24	Node name : raster-extractions
2025-10-17 15:10:24	Node name : wdp
2025-10-17 15:10:24	Node name : timeseries-validation
2025-10-17 15:10:24	Node name : tsmdispatcher
2025-10-17 15:10:24	Node name : dispatcherdb
2025-10-17 15:10:24	Node name : queueserver
2025-10-17 15:10:24	Node name : automation
2025-10-17 15:10:24	Node name : interactive
2025-10-17 15:10:24	Node name : scriptserver
2025-10-17 15:10:24	Node name : tsml
2025-10-17 15:10:24	Node name : karaf-node0
2025-10-17 15:10:24	Node name : processanalytics-importer
2025-10-17 15:10:24	Node name : python-fastapi-runner
2025-10-17 15:10:24	Node name : thirdparty-node-red
2025-10-17 15:10:24	Node name : water-python-components
2025-10-17 15:10:24	Node name : water-workflow-engine
2025-10-17 15:10:28	C:\KISTERS\wiski\services\config\local\unmanaged\security\keystore.p12

7 Initial Installation Using the WSM (Empty Database)

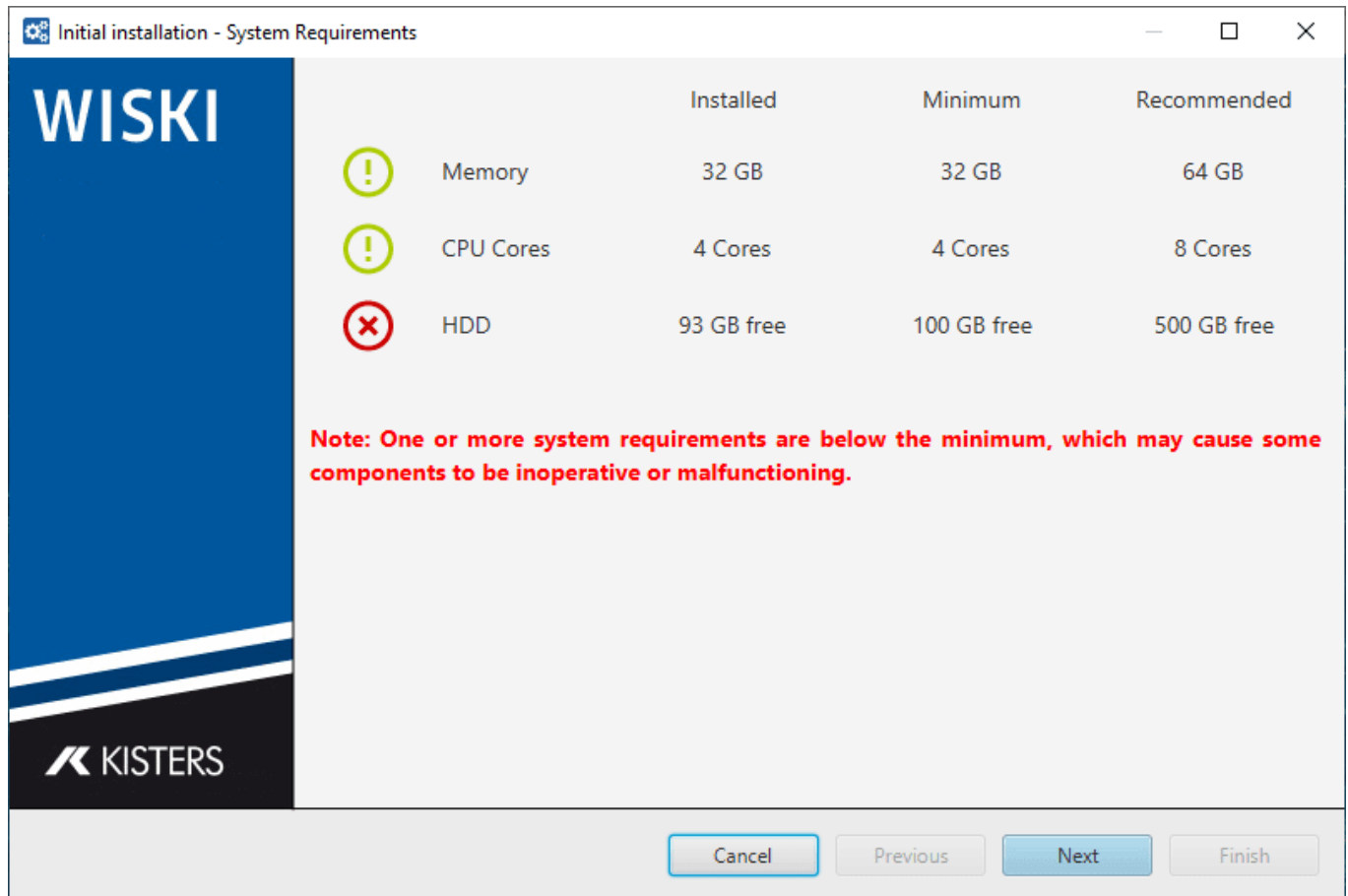
Put together a [package configuration](#). If, for example, you would like to install a 64-bit WISKI server installation, you need the following packages:

- WISKI-<vn>-<bn>-java-64bit.zip
- WISKI-<vn>-<bn>-jprocontroller-client-64bit.zip
- WISKI-<vn>-<bn>-jprocontroller-client-default-config.zip
- WISKI-<vn>-<bn>-kidsm.zip
- WISKI-<vn>-<bn>-kidsm-client-64.zip
- WISKI-<vn>-<bn>-kilog-client-64bit.zip
- WISKI-<vn>-<bn>-services.zip
- WISKI-<vn>-<bn>-wiski-client.zip
- WISKI-<vn>-<bn>-wiski-client-help-all.zip

In the **WIZARDS** tab of the WSM main screen, click the **[INITIAL INSTALLATION]** button.



The software checks if your system meets the system requirements. Click the **[NEXT]** button to dismiss this dialog.



WISKI

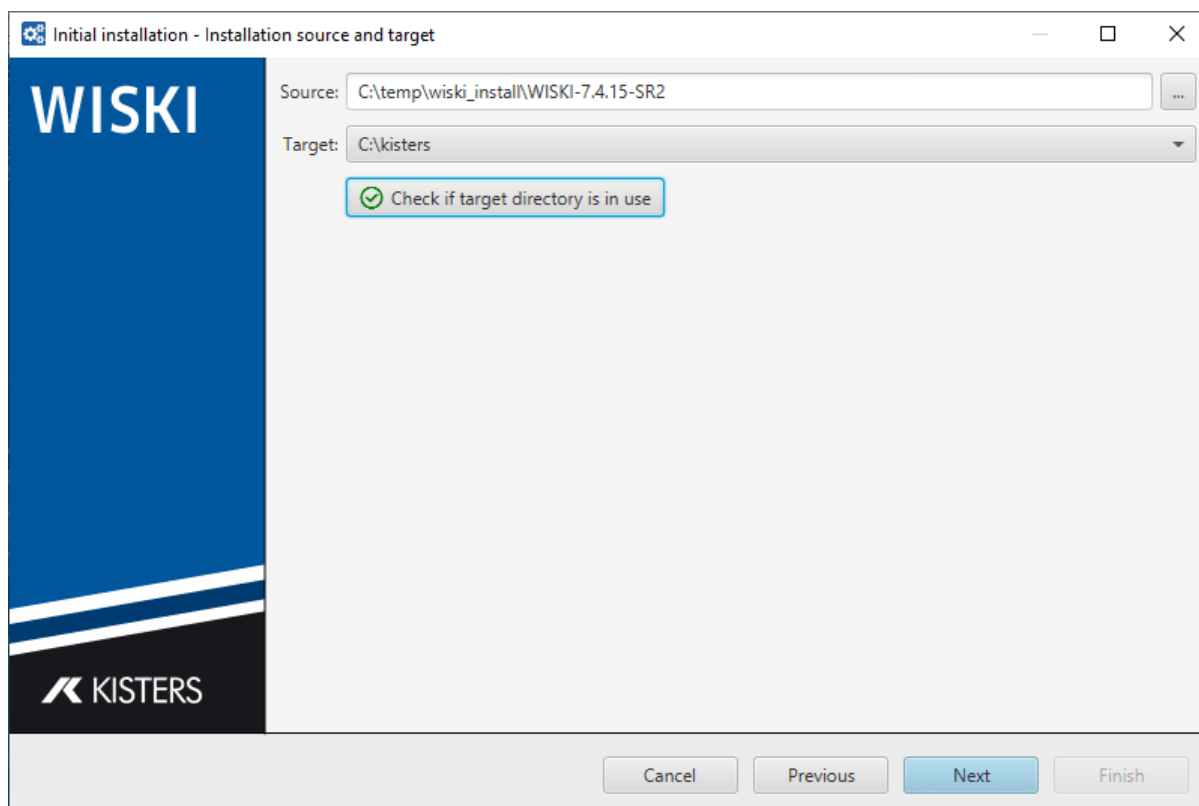
	Installed	Minimum	Recommended
Memory	32 GB	32 GB	64 GB
CPU Cores	4 Cores	4 Cores	8 Cores
HDD	93 GB free	100 GB free	500 GB free

Note: One or more system requirements are below the minimum, which may cause some components to be inoperative or malfunctioning.

KISTERS

Buttons: Cancel, Previous, Next, Finish

The following step is to select the source directory. This folder must contain the installation packages.



WISKI

Source: C:\temp\wiski_install\WISKI-7.4.15-SR2

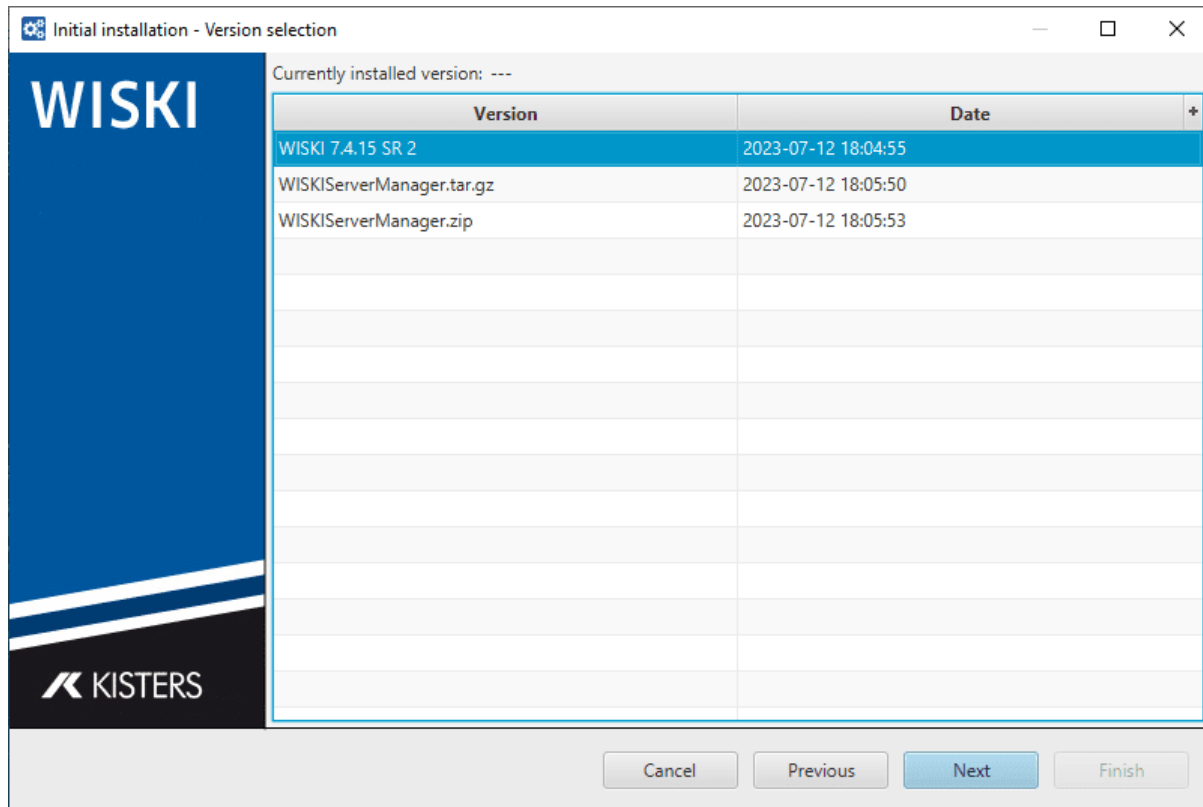
Target: C:\kisters

☒ Check if target directory is in use

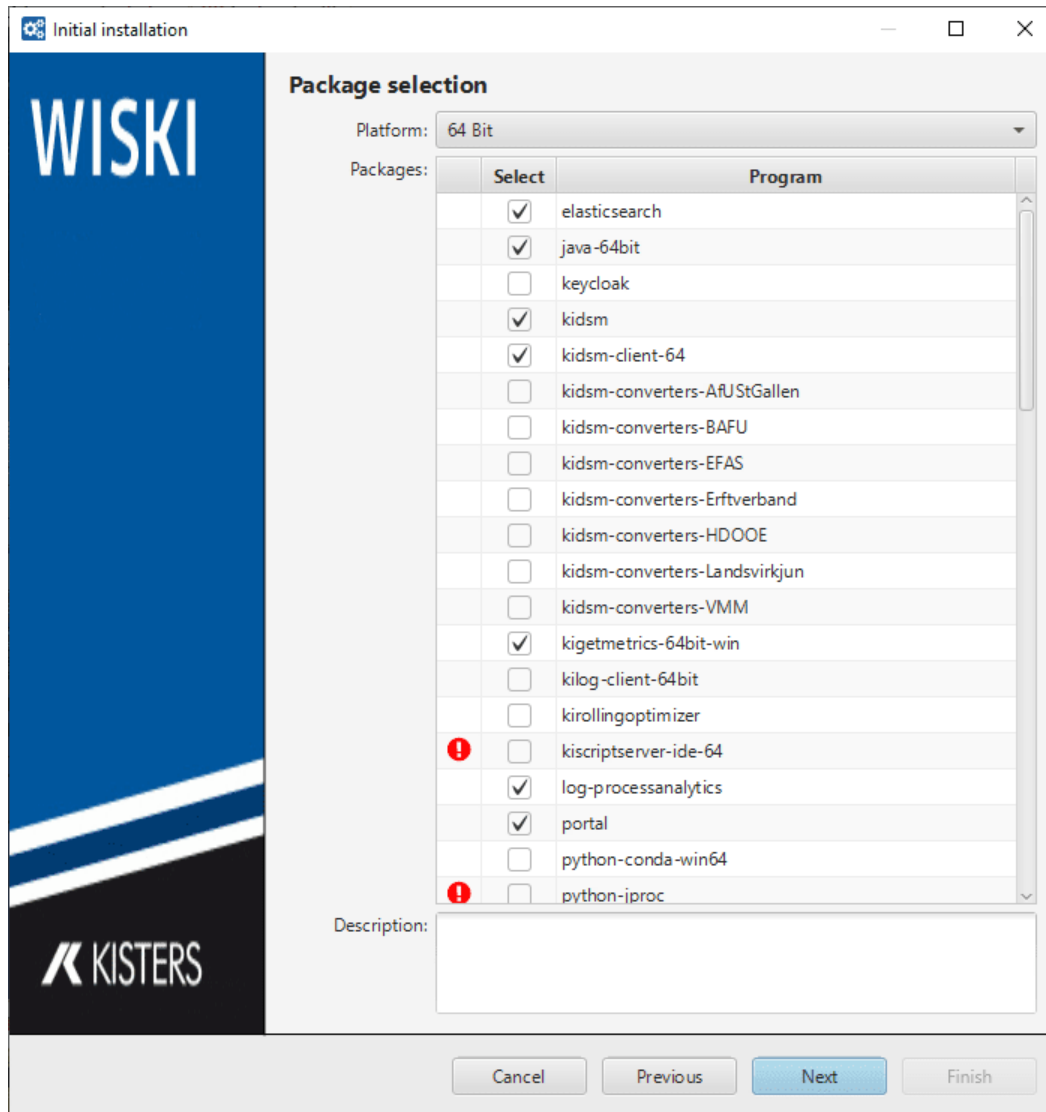
KISTERS

Buttons: Cancel, Previous, Next, Finish

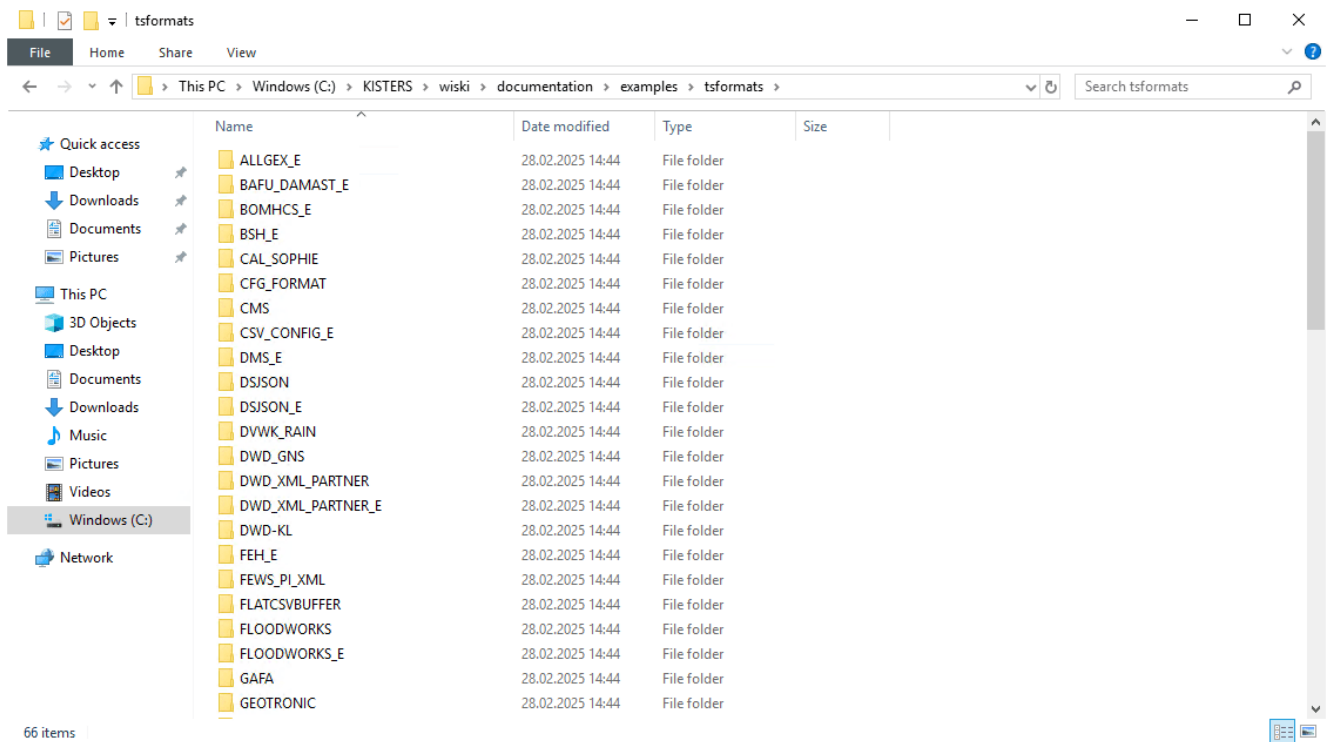
In the next dialog you can select the desired version of WISKI7. The available displayed versions are found in the directory that was specified in **SOURCE** earlier.



The next step is to select the packages which are to be installed.

**Note:**

- In the **PACKAGE SELECTION** step you will only see the packages that are necessary for an operational WISKI7 system or for which you have purchased additional licenses. Ask your project representative for details.
- You can optionally select a WSM package with examples for time series import/export formats. The format examples are saved to the `<root>\documentation\examples\tsformats` subfolder.



Package selection

WISKI Client/Server

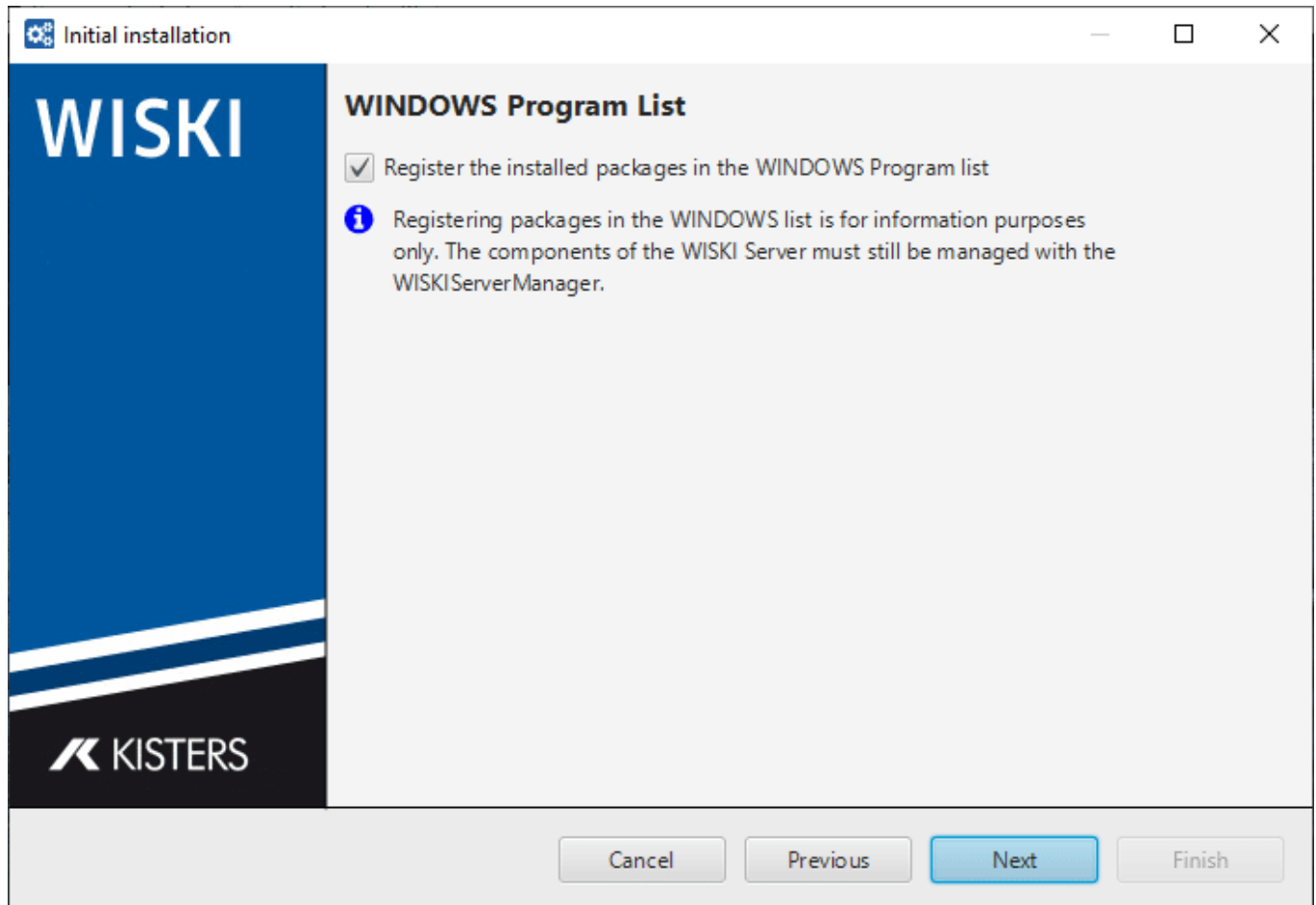
- Install the **elasticsearch** package if you want to use [Elasticsearch](#)^[223]. This package is also required when working with the Data Harvester.
- For more information about the Data Harvester please contact your KISTERS representative.
- Install the **kigetmetrics-64-win** package to use the WISKI metrics in the WISKI client.
- Install the **kirollingoptimizer** package to use the special software KiRollingOptimizer (ask your project representative for details).
- Install the **water-python-win64 (-linux64)** and **python-jproc** package to use the Python Framework for Reporting and Operation Framework.
 - Additionally you need to install your customer specific packages (customer report bundle) eg. **python-customer-<CUSTOMER>-win64**, **python-reporting-<CUSTOMER>-win64** to use your customer specific reports and operations (ask your project representative for details).
- Install the **water-workflow-engine** package to use the WISKI workflows.
- Install the **water-io-wiski-client-help-examplefiles** package with examples for time series import/export formats mentioned above if you want to use example files during your import or export work.
- Install the package **KiDBTUtil** for the standalone time series export transfer tool that reads data from various JDBC-capable databases and imports the data to WISKI (TSM). See the corresponding appendix in the WISKI user documentation.

WISKI Portal (Web)

- Install the **portal** package to work with the [WISKI Standard Portal](#)^[181].
- Install the **elasticsearch** package if you want to use [Elasticsearch](#)^[223] and you do not have an Elasticsearch database. Elasticsearch is required when working with the Web application [Process Analytics](#)^[197].
- Install the **log-processanalytics** package to use the Web application [Process Analytics](#)^[197].
- Install the **kigetmetrics-64-win** package to use the Web application [Application Metrics](#)^[195].
- Install the **wdp-win** and **kiwis** package for **Portal installations after 7.4.11 SR7**.
- Install the **python-jproc**, **validation-karaf**, **validation-python**, **water-workflow-engine** and **water-python-win64** package to use the Web application [Data Validation](#)^[203].
- Install the **kidsm** package to use [KiDSM](#)^[200].

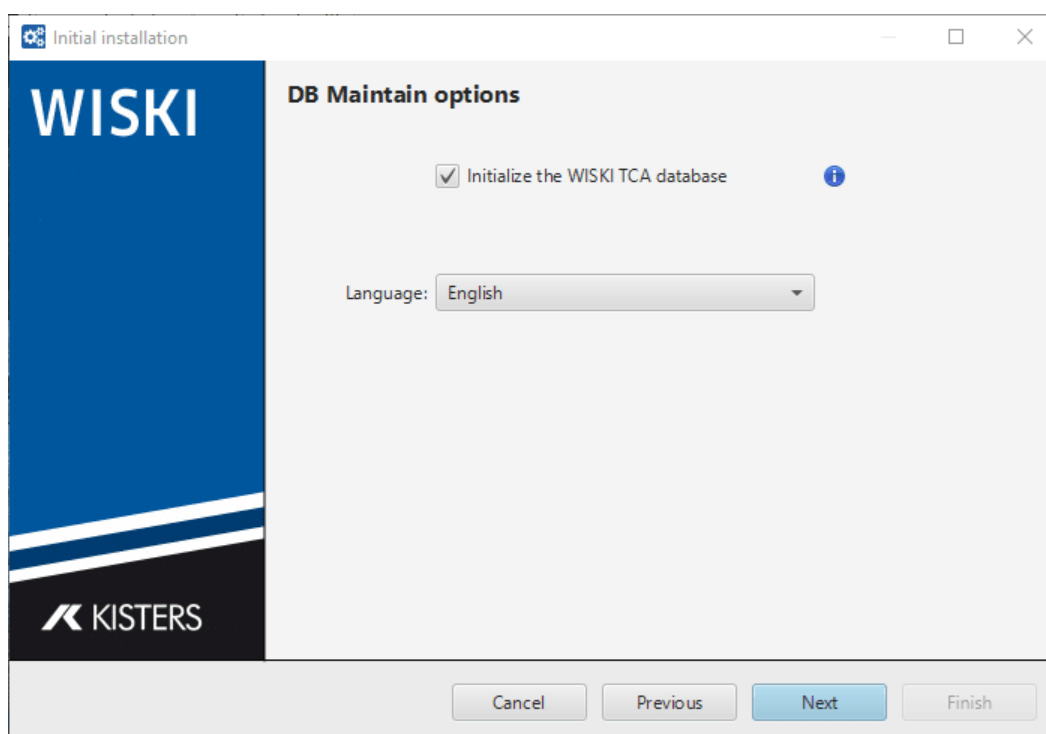
During the installation, several database initialisation programs are executed. If you want to run one of these steps manually at a later point, the corresponding scripts can be deselected here by unchecking the respective box.

In this dialog you choose if the WISKI7 installation packages should be listed in the Windows Programs and Features list.



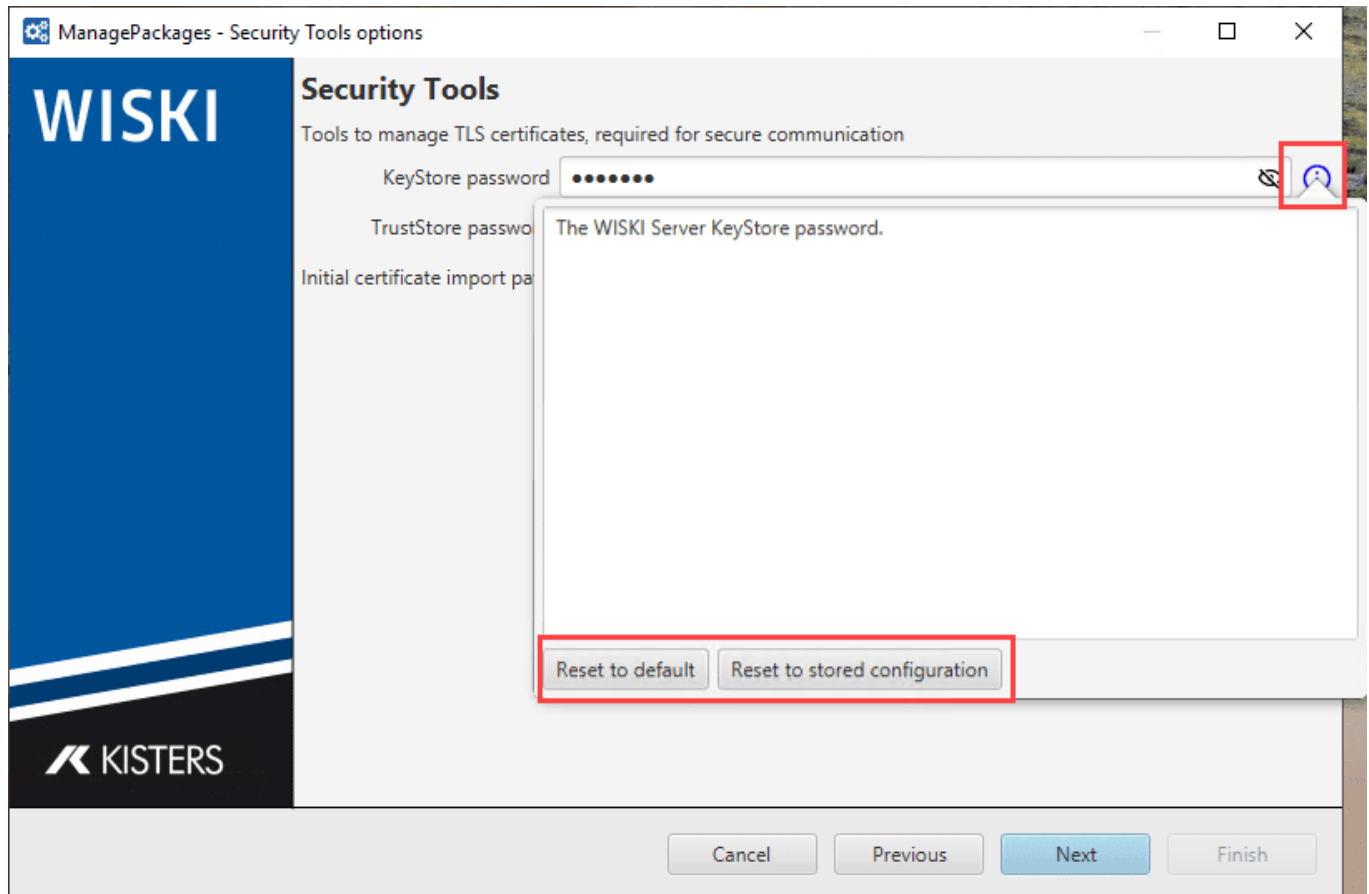
The WISKI database contains multi-language data. Therefore, an installation language needs to be selected in this step.

Warning: The installation language cannot be changed after the initial installation.

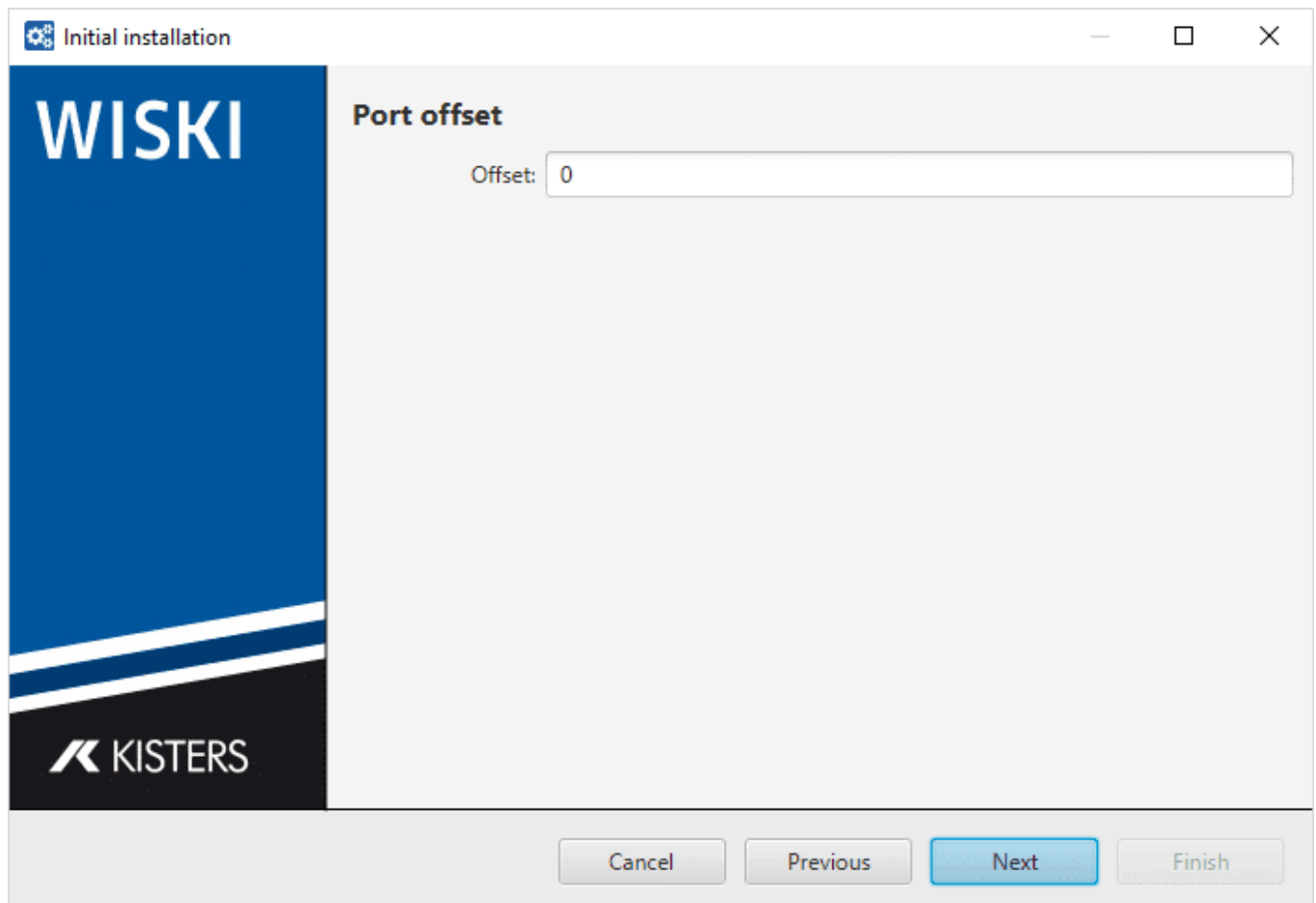


Note: There are additional tooltips in each dialog that can be found by clicking on the information icon .

In some contexts this icon also offers functions to execute, e.g. restoring defaults or similar. See this example:



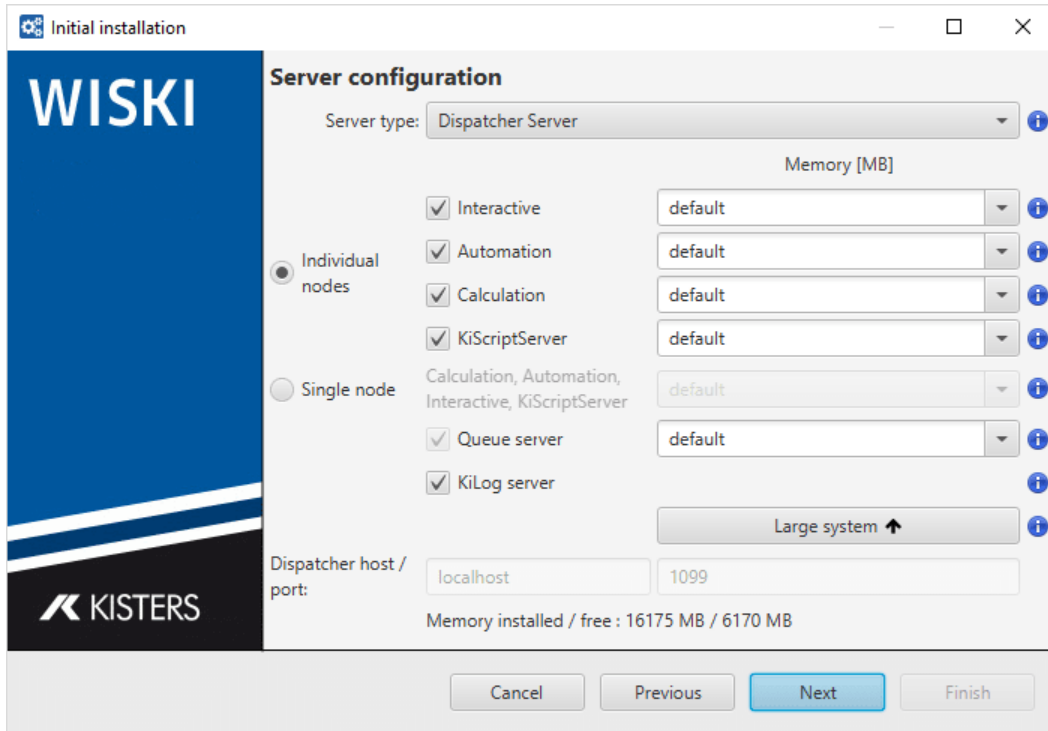
The WISKI installation uses several ports (see [the list of available ports](#)^[14] for details, as well as the [ports communication matrix](#)^[25]). If you want to install two or more systems, the port range can be shifted by an offset.



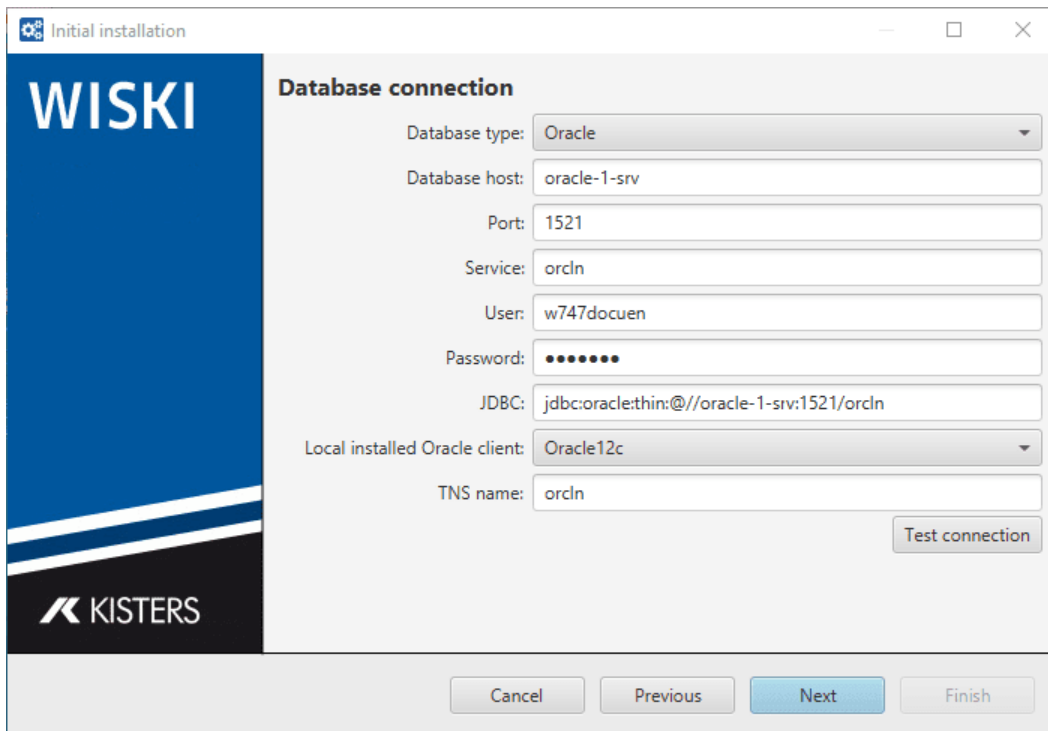
See [Port Overview](#)  for ports and their configuration.

In the next window the server configuration can be specified.

You can configure all nodes of your server individually or use the **SINGLE NODE** option for **INTERACTIVE**, **AUTOMATION**, **KIScript** and **CALCULATION**. This will reduce the number of nodes in your WSM Service Monitor.



The next step is to enter the database connection information. This will be used to connect to the database. Before the **[NEXT]** button is activated, the connection must be tested. You can do this by clicking on **[TEST CONNECTION]**.



In the next steps of the installation or update process, you will have to enter your credentials for the Keycloak database connection.

Note: We recommend using your own database schema for Keycloak. It is possible to install Keycloak inside the WISKI schema, but we do not recommend this for security reasons.

Initial installation - Database connection

WISKI

Keycloak database connection

Note : It is recommended to use a separate database schema for Keycloak. If thats not possible, you can also use the same schema than WISKI Server.

Database type:

Database host:

Port:

Service:

User:

Password:

JDBC:

The next window asks for the Keycloak admin password. Note that the Keycloak admin user is named **keycloak** and its default password is **12years**.

Optionally, the configuration of the LDAP backend can be enabled by clicking **USE LDAP**.

Fill in your LDAP credentials.

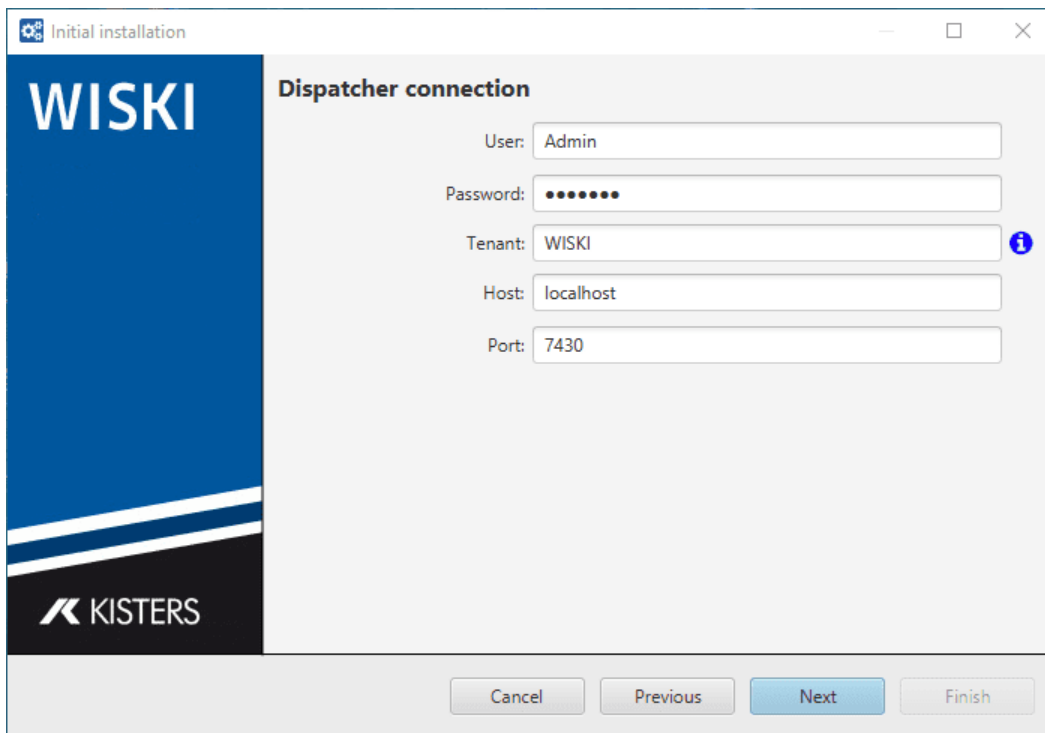
The next window contains the optional LDAP configuration. Fill in your LDAP credentials, if required.

The table below explains the single elements and settings.

Element	Description
Keycloak password admin	Enter your Keycloak admin password; the Keycloak admin user by default is keycloak and its default password is 12years .
Keycloak password keystore	
Use LDAP	Optionally, the configuration of the LDAP backend can be enabled by checking this box.
LDAP URL	The URL of the machine where the LDAP server is running.
Bind DN	The distinguished name of the connection parameters to LDAP including the necessary rights, consisting of common name (CN), organisation unit (OU) and domain controller (DC).
Bind Credential	The password of the LDAP user.
Users DN	Full Distinguished Name of LDAP tree where your users are. This DN is parent of LDAP users. It could be for example 'ou=users,dc=example,dc=com' assuming that your typical user will have DN like 'uid=john,ou=users,dc=example,dc=c [...]'; the distinguished name of the current user on LDAP; this indicates how the users are organised on the LDAP server, with the organisation units (OU) and the domain controller(s) (DC).
LDAP backend type	Select one of the options to define the LDAP type. This indicates if your LDAP backend is an Active Directory or not (select OTHER).
[Test connection]	Click this button to test the connection between the WISKI server and the LDAP server.

See [WISKI Open ID Connect Configuration](#)^[134] for more details on using Keycloak.

In the next step, the Dispatcher login can be specified. For the initial installation the defaults should be maintained.

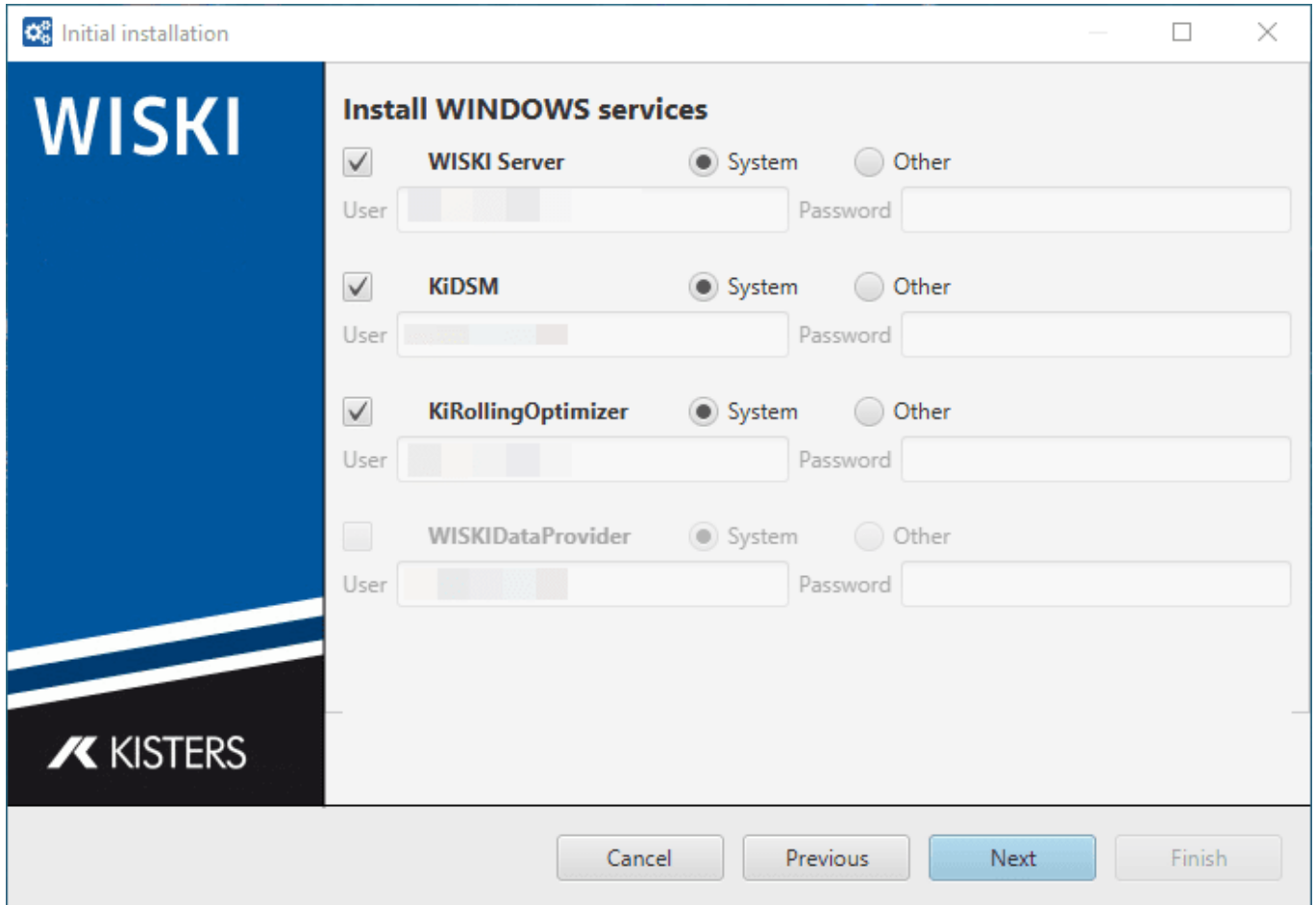


The screenshot shows a window titled "Initial installation" with the WISKI logo on the left and the "Dispatcher connection" configuration form on the right. The form contains the following fields:

- User: Admin
- Password: (masked with dots)
- Tenant: WISKI
- Host: localhost
- Port: 7430

At the bottom of the window, there are four buttons: "Cancel", "Previous", "Next", and "Finish". The "Next" button is highlighted in blue.

In the next dialog you can specify the Windows services to be installed and the user running it.



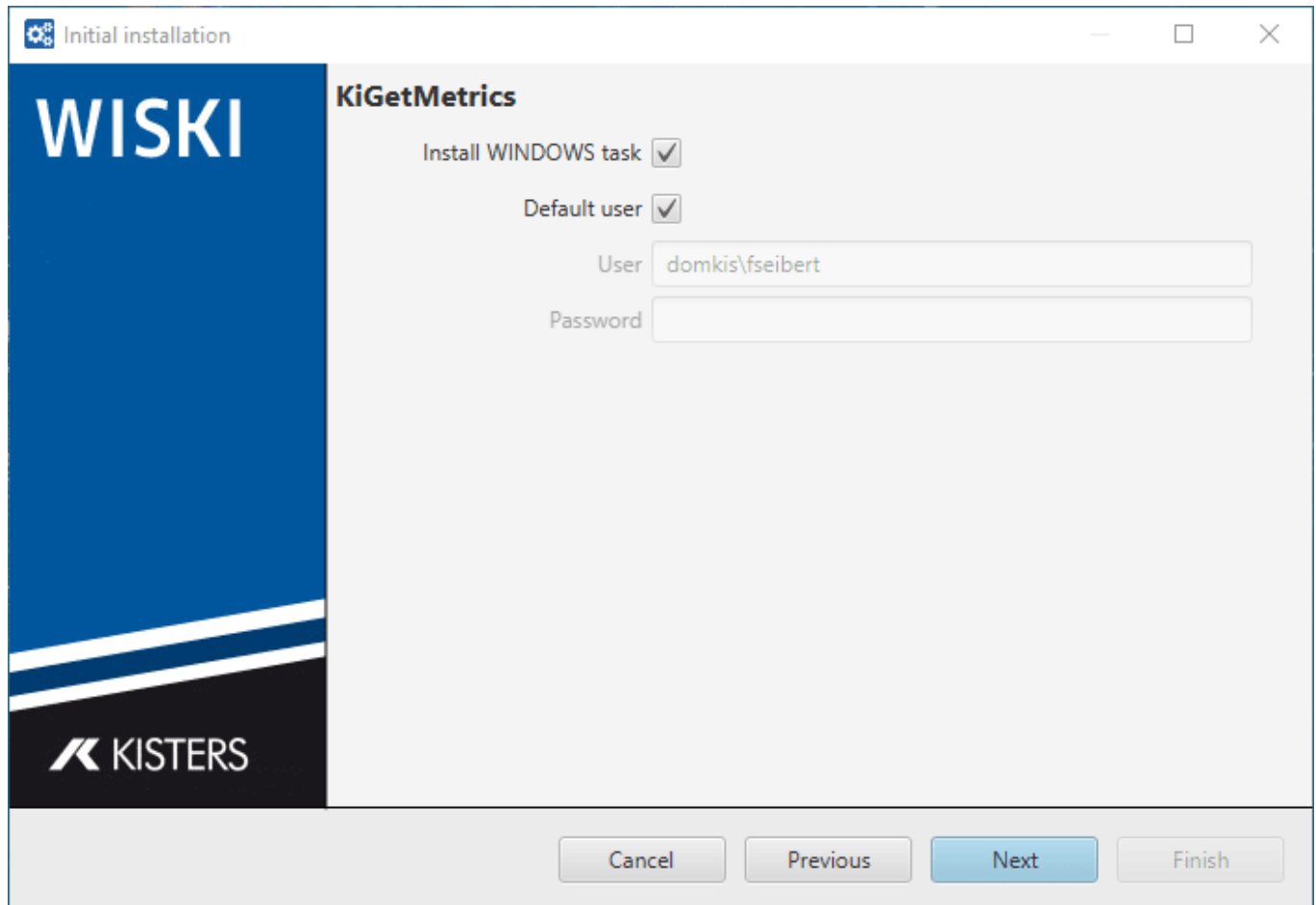
The image shows a Windows installer window titled "Initial installation". On the left is a blue sidebar with the "WISKI" logo at the top and the "KISTERS" logo at the bottom. The main area is titled "Install WINDOWS services" and contains four service entries, each with a checkbox, a name, radio buttons for "System" or "Other", and fields for "User" and "Password".

Service	Selected	Run as	User	Password
WISKI Server	☒	☒ System ☐ Other	<input type="text"/>	<input type="password"/>
KiDSM	☒	☒ System ☐ Other	<input type="text"/>	<input type="password"/>
KiRollingOptimizer	☒	☒ System ☐ Other	<input type="text"/>	<input type="password"/>
WISKIDataProvider	☐	☒ System ☐ Other	<input type="text"/>	<input type="password"/>

At the bottom right are four buttons: "Cancel", "Previous", "Next" (highlighted in blue), and "Finish".

In the next window you can define whether or not the KiGetMetrics Windows task is installed and the user running it.

Note: This window only appears if you have selected the **KIGETMETRICS** package in the **PACKAGE SELECTION** dialog.



The image shows a Windows-style installation window titled "Initial installation". On the left is a blue and black sidebar with the "WISKI" logo in white and the "KISTERS" logo at the bottom. The main area is titled "KiGetMetrics" and contains two checked checkboxes: "Install WINDOWS task" and "Default user". Below these are two text input fields: "User" with the text "domkis\fseibert" and an empty "Password" field. At the bottom right are four buttons: "Cancel", "Previous", "Next" (highlighted in blue), and "Finish".

Initial installation

WISKI

KiGetMetrics

Install WINDOWS task ☒

Default user ☒

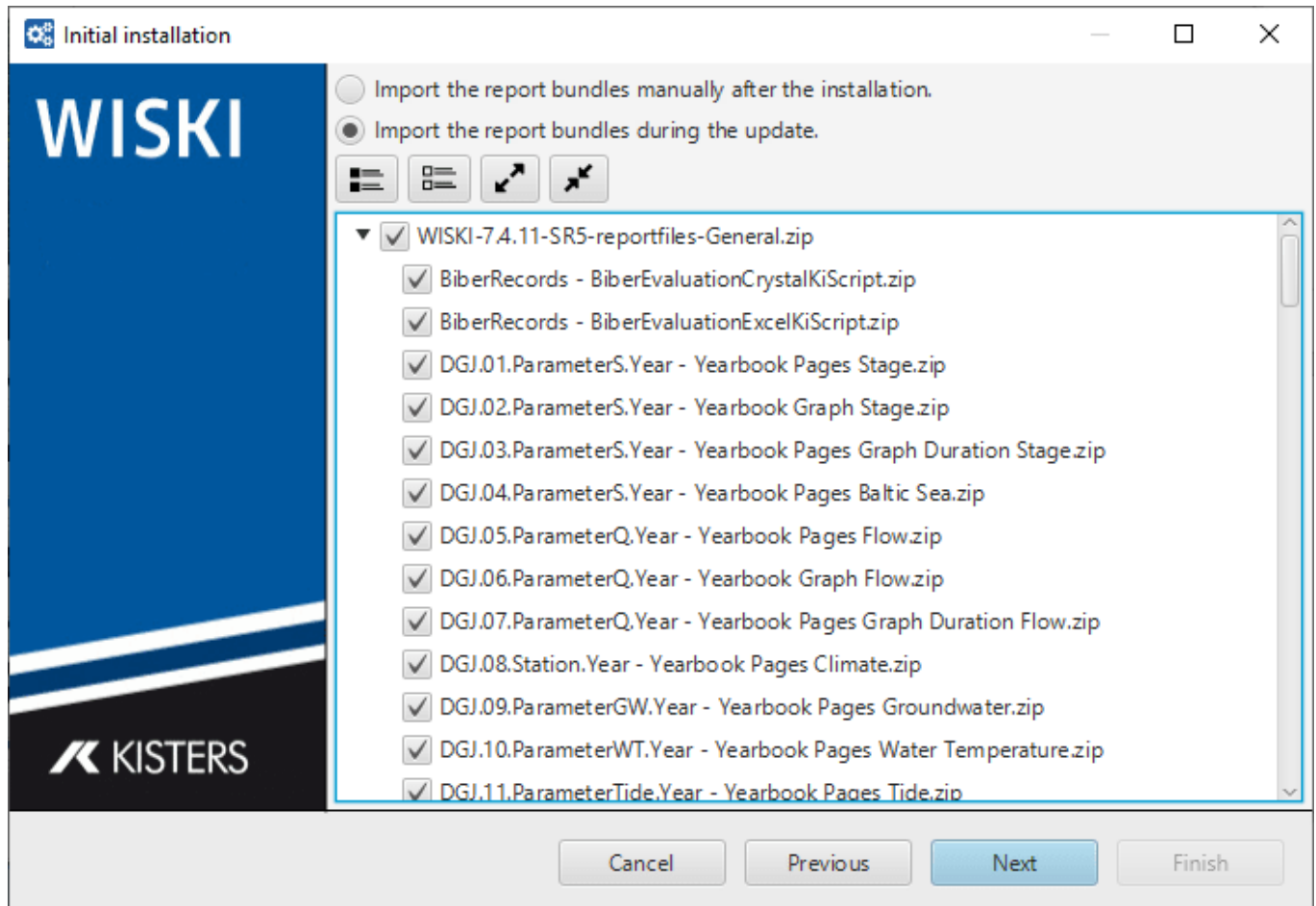
User

Password

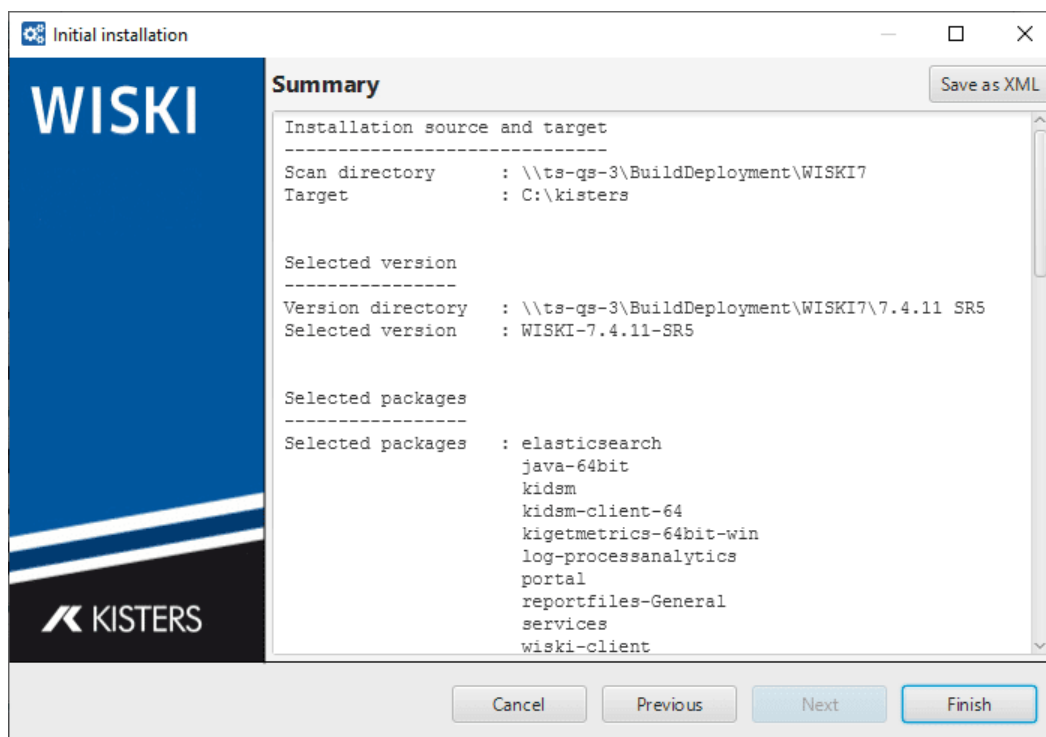
KISTERS

Cancel Previous Next Finish

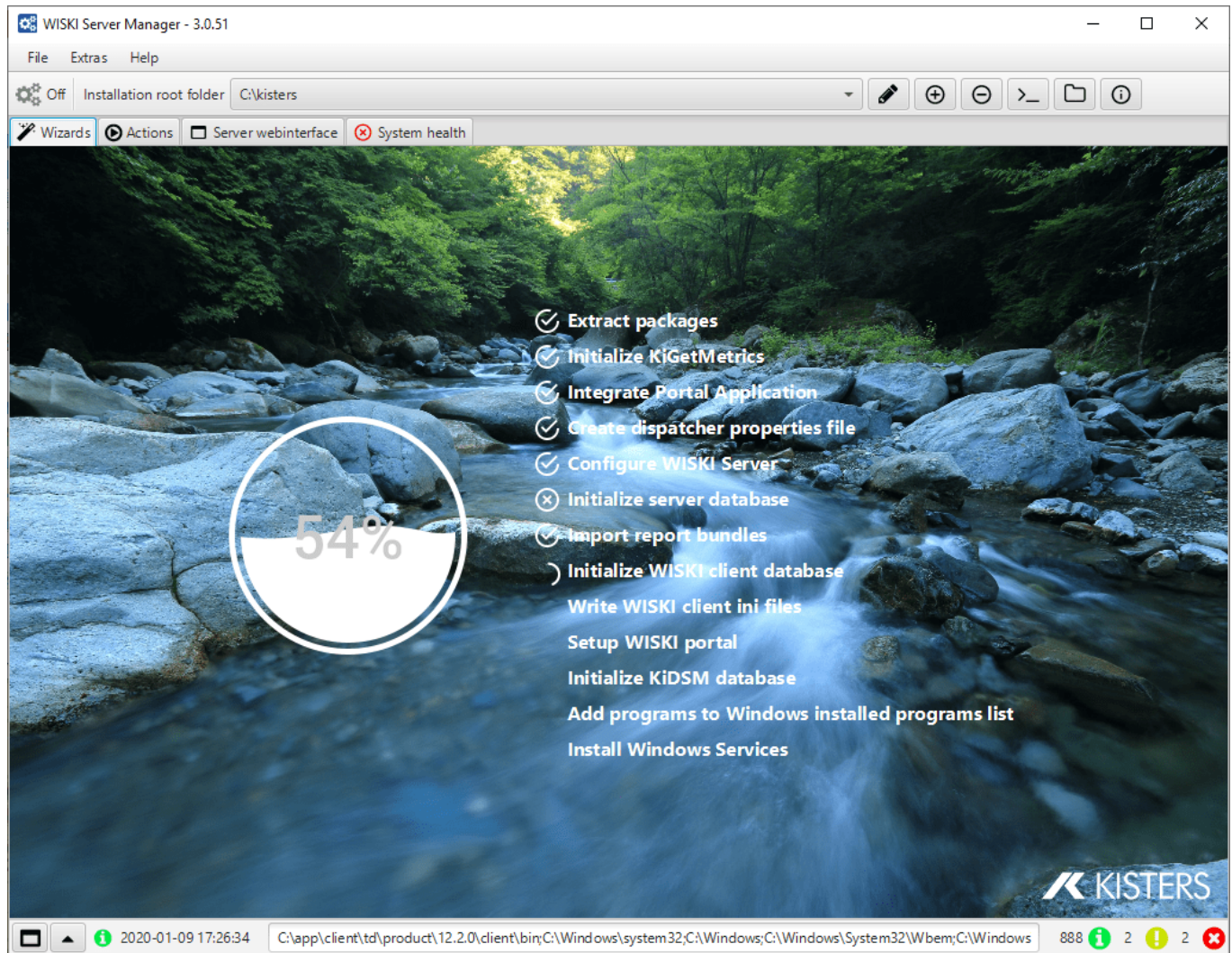
The next step is to select the report bundles to be imported. Alternatively, you can import the report bundles after the installation.



When clicking the **[FINISH]** button, the installation is executed.



All steps are logged in the following window.



8 Updating an Installation Using the WSM (Existing Database)

! CAUTION

Before updating a WISKI server installation, be sure to perform the [Pre-Update Checks](#).

The WISKI Server Manager is designed to update the complete installation, including all client and server components.

The user specific configuration files (like `wiski*.ini`, `DbMaintain*.ini`) are saved and copied back inside the update process. In case of special customer specific map files, use the pattern `-custom` (like `extended-custom.map`) to identify the files to be stored in the backup as well.

By executing the action **CONFIGURATION - BACKUP THE SYSTEM CONFIGURATION** you can see which files are saved into the backup folder.

In order to update the software installation, the WISKI Server Manager needs full control over the `<root>\wiski` directory. To get full control, all services need to be shut down and uninstalled (see ch. [Windows services](#) ^[206]). All clients need to be closed, including all Windows Explorers pointing to one of the subdirectories.

Note: If the previous installation is a version lower than WISKI 7.3, make sure that your software `<root>` directory does not contain a `wiski` subdirectory.

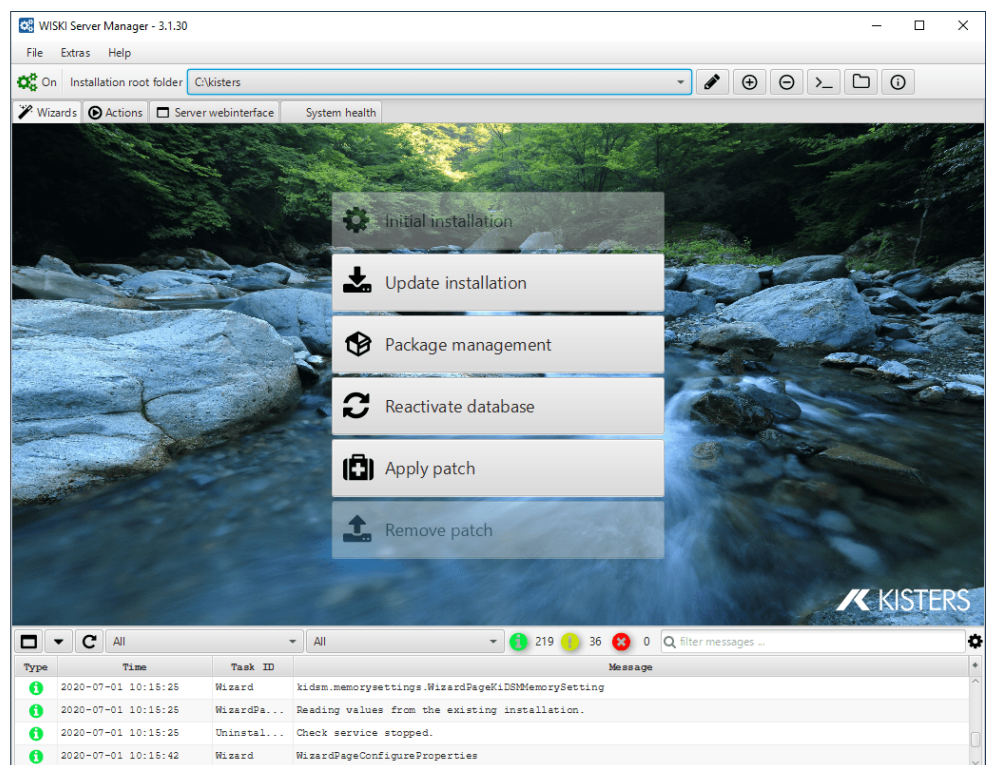
Perform the following steps to do the update:

In the **WIZARDS** tab, click the **[UPDATE INSTALLATION]** button.

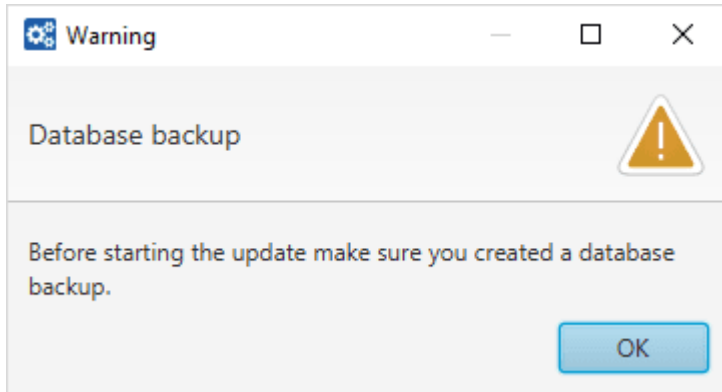
Currently selected
software location

Available wizards

Logging area



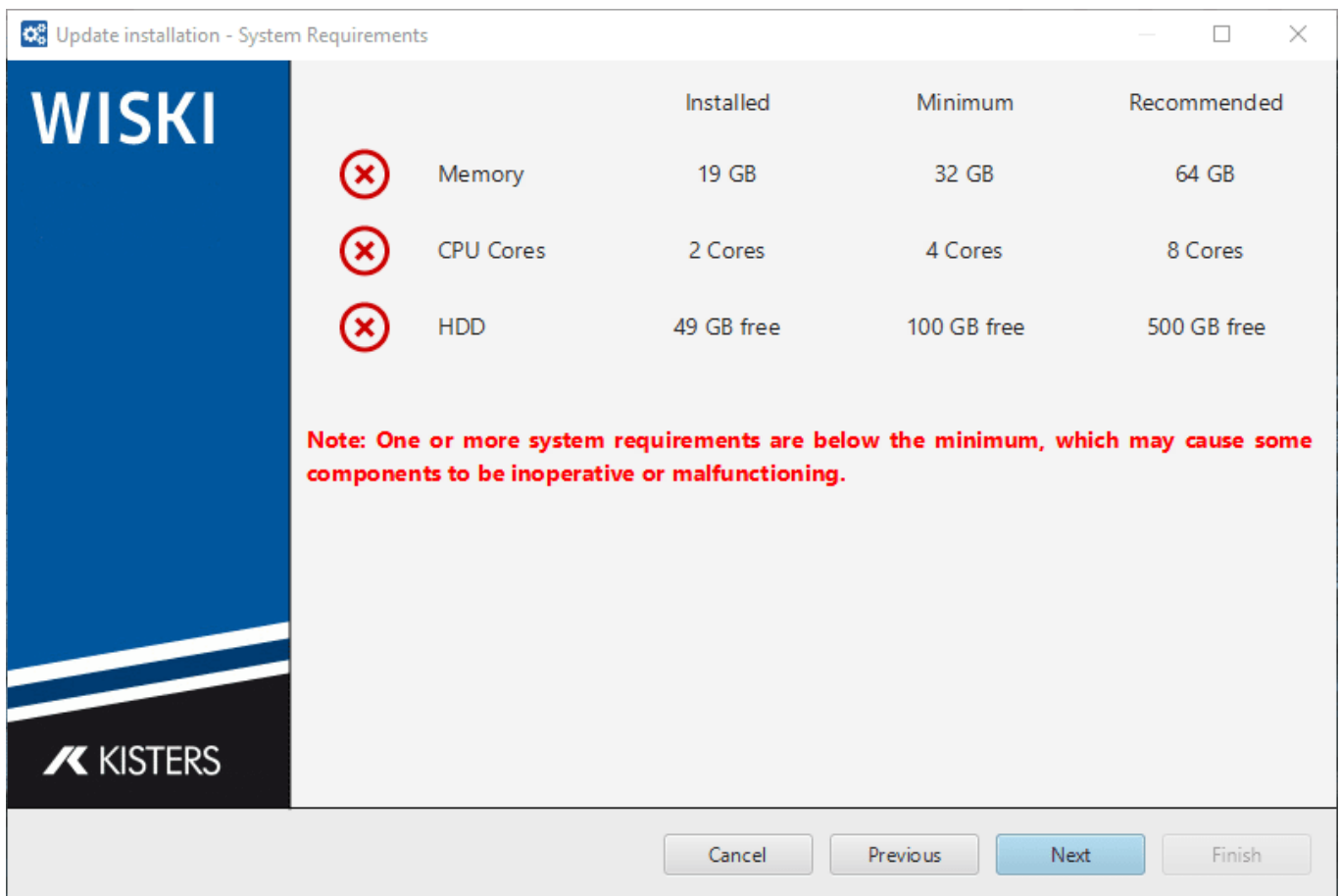
You get a system message:



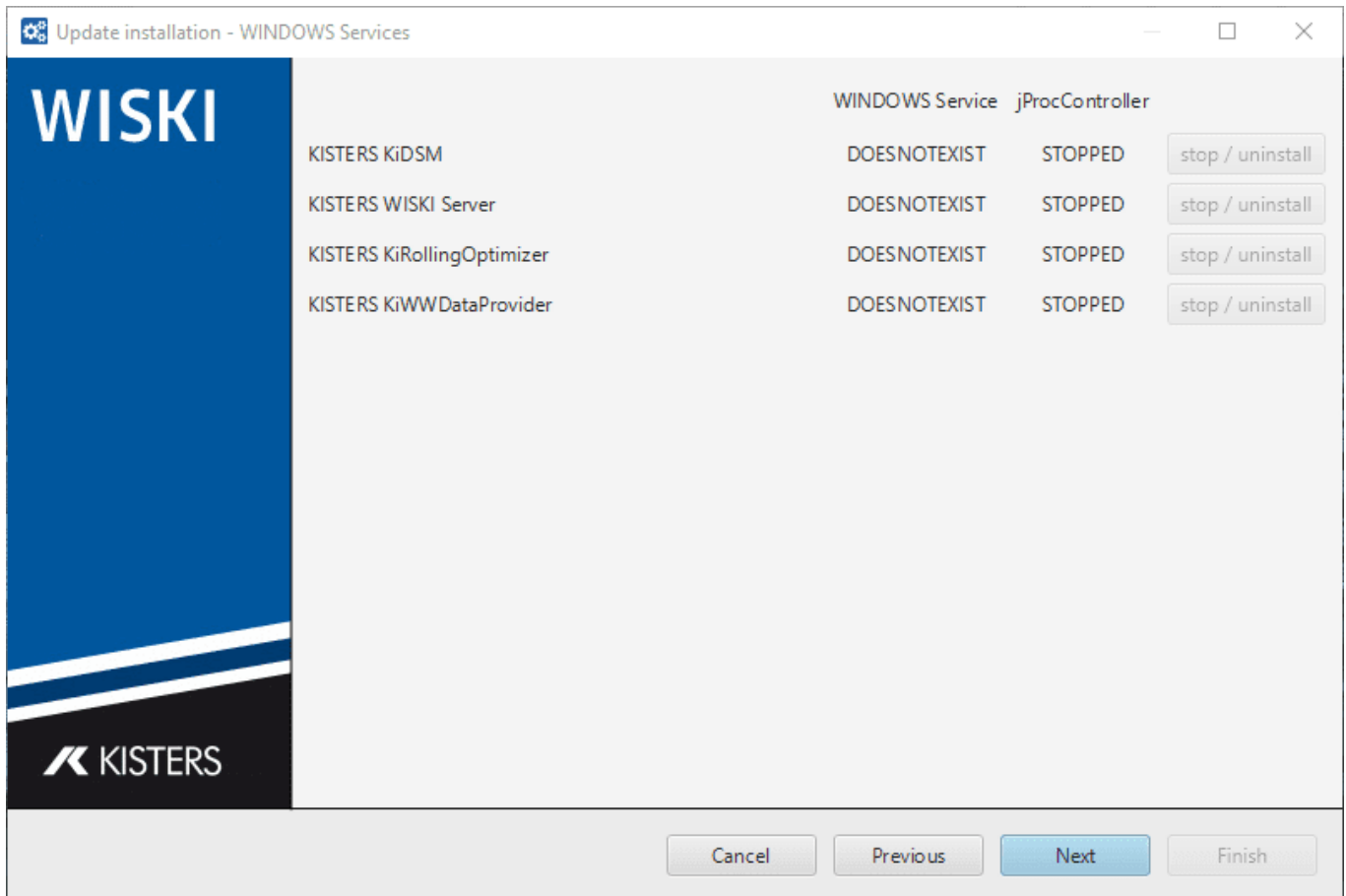
Backup your database for security reasons!

Confirm with **[OK]**.

The software checks if your system meets the system requirements. Click the **[NEXT]** button to dismiss this dialog.

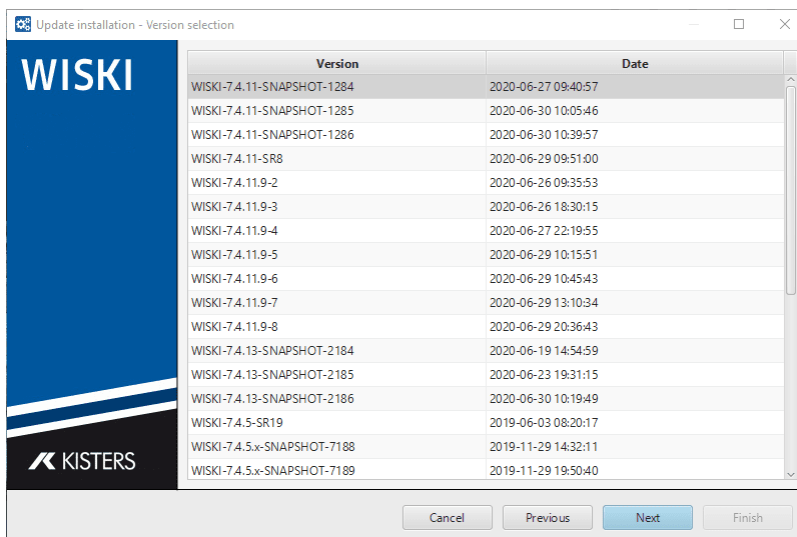


In the following step the updater checks for active WISKI related services and processes. Furthermore you have the option to stop or uninstall certain services.

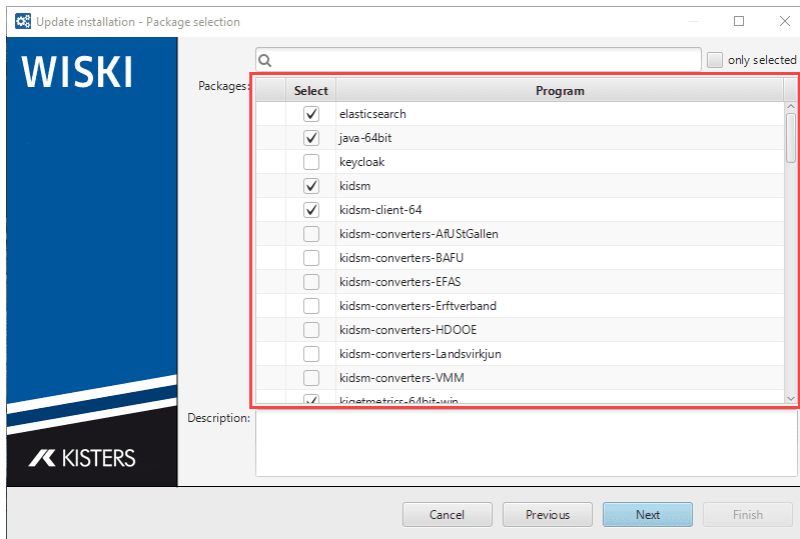


Use the dialog **INSTALLATION SOURCE AND TARGET** to select the source folder and make sure that the WISKI directory is not in use. The source folder must contain the version which is to be installed. Make sure to stop the WISKI server before updating as well.

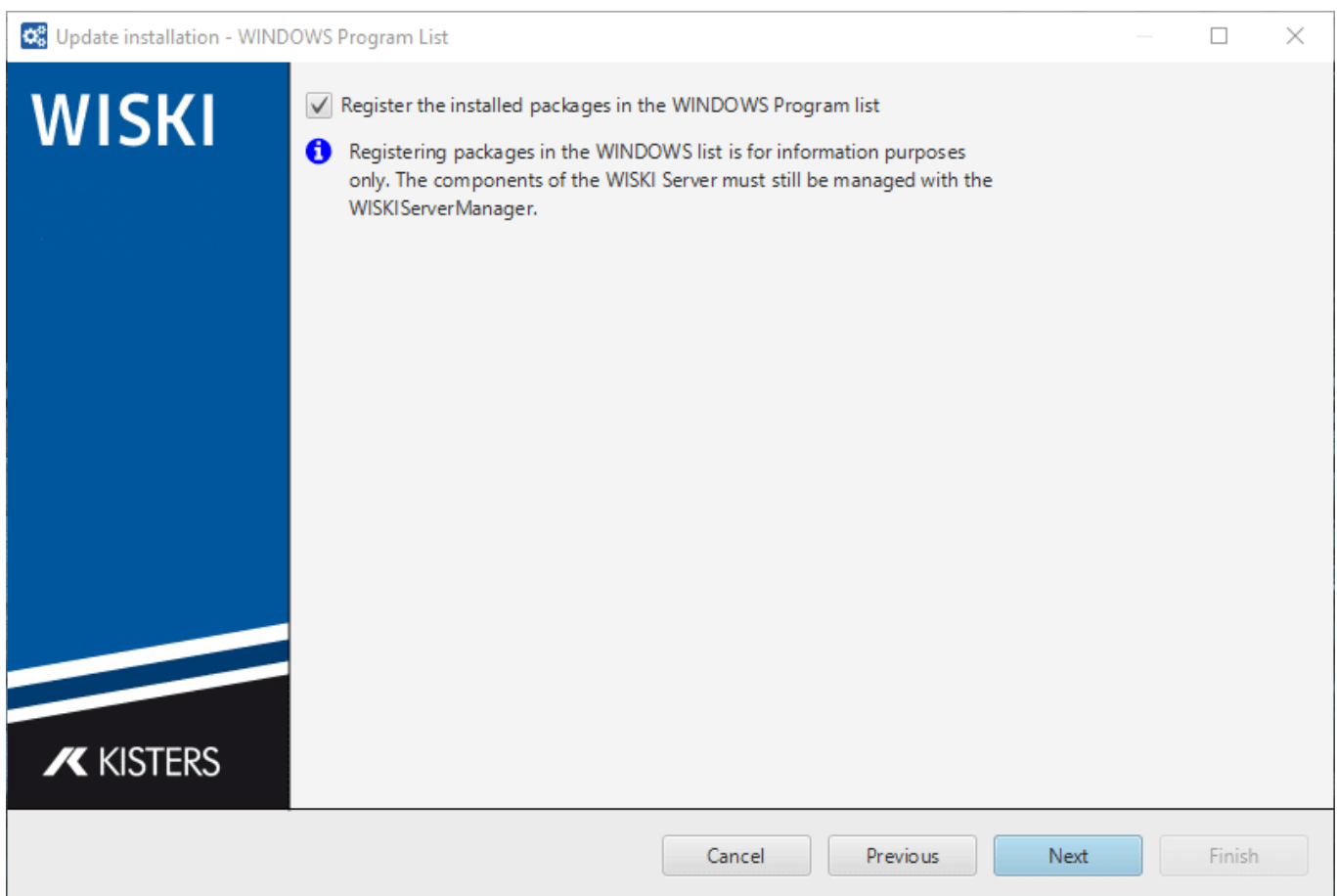
In the next window you need to select the desired version on which you like to update.



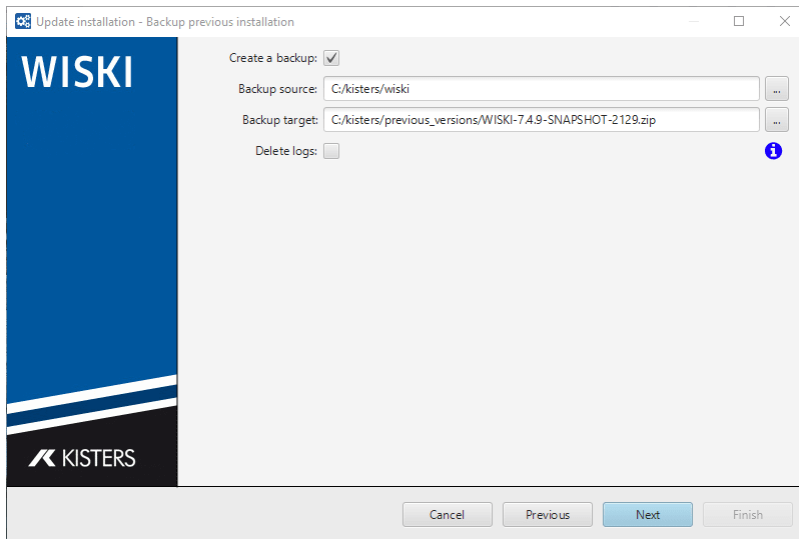
The next step is to select the packages which are to be installed. By default, the packages that were selected during the last installation are pre-selected.



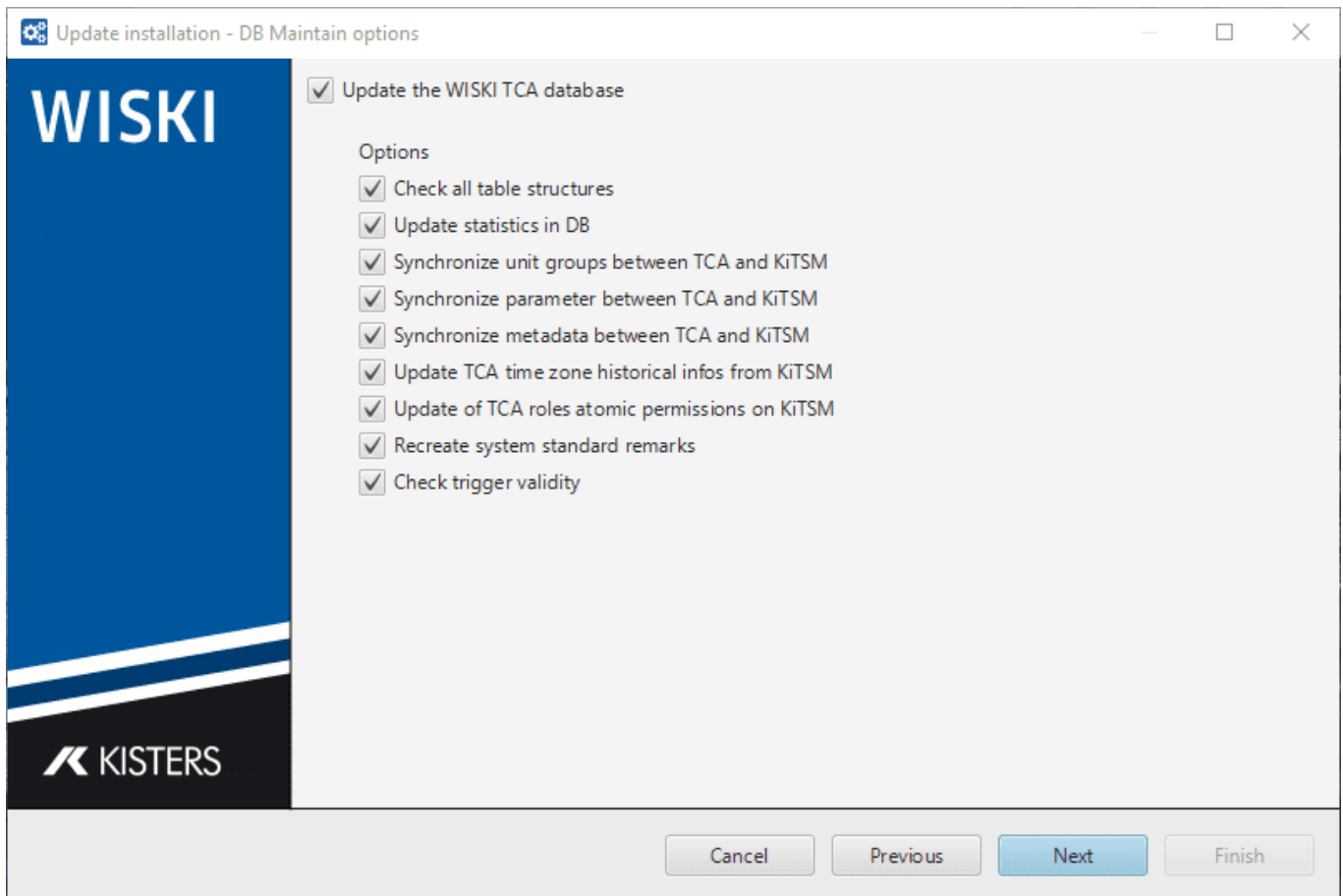
In this dialog you can choose if the WISKI7 installation packages should be listed in the Windows Programs and Features list.



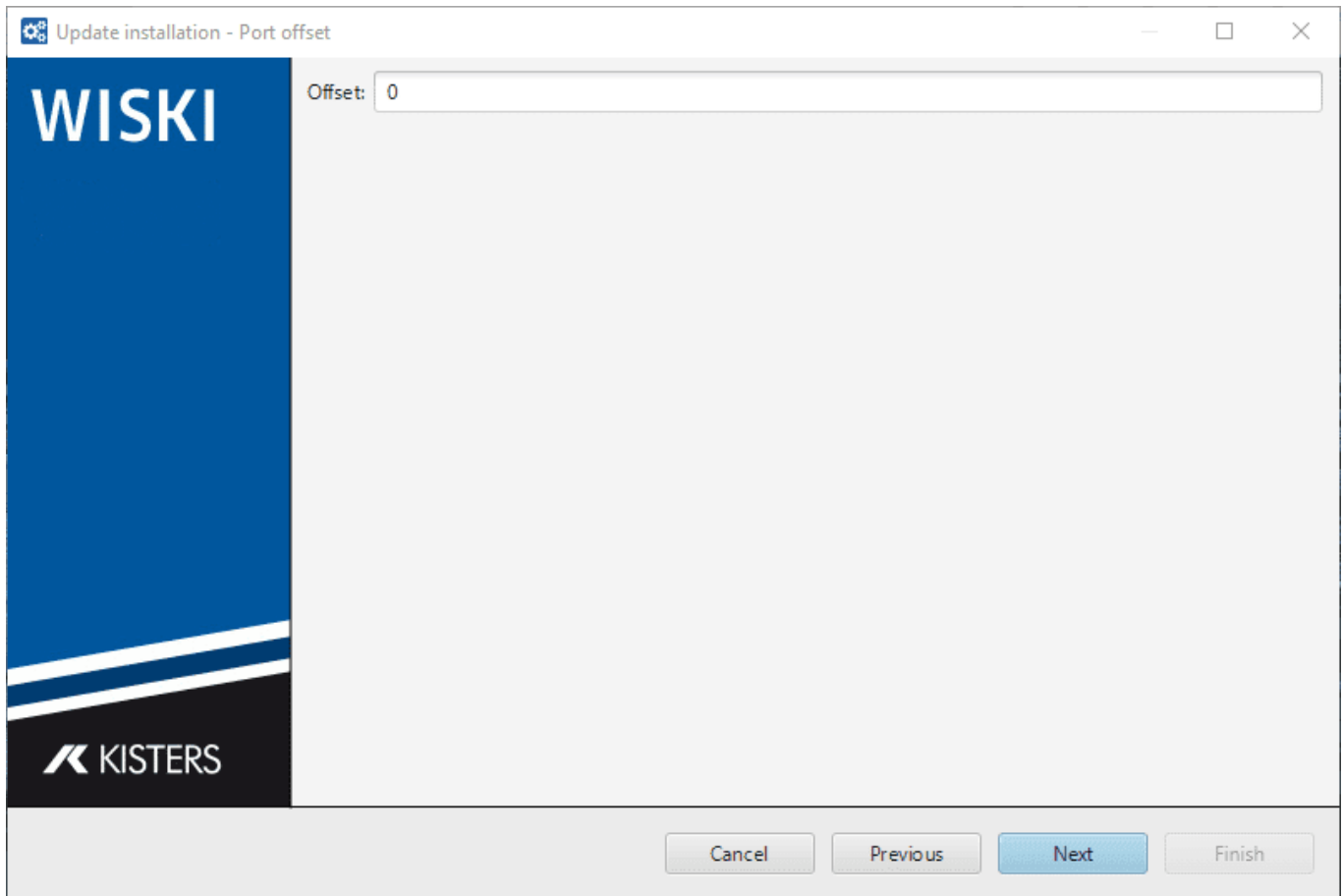
During the update procedure, the complete existing software installation folder is deleted. Therefore, it is recommended to create a backup of the to-be-deleted directory in the next step. If the directory with your current installation is found by the installer, a backup file name will be entered automatically in the wizard page. By default, the backup file is stored in `<root>\previous_versions`.



In the next step you can specify maintenance options like known from **DbMaintain.exe**. For further information on database maintenance, please refer to chapter [Initialising the WISKI7 Database](#) ¹⁰⁴.



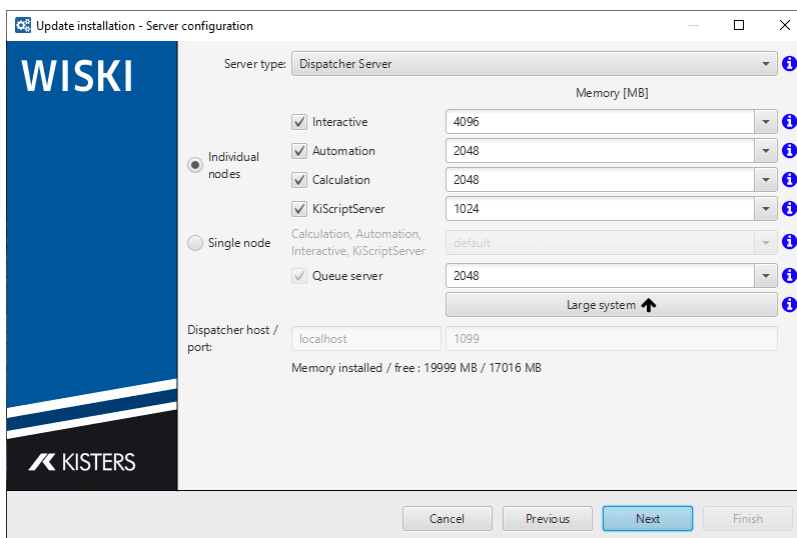
In the next step during an update, you need to change the port offset if necessary. The last selected value of the previous installation is displayed here by default.



The following dialog lets you change the server configuration.

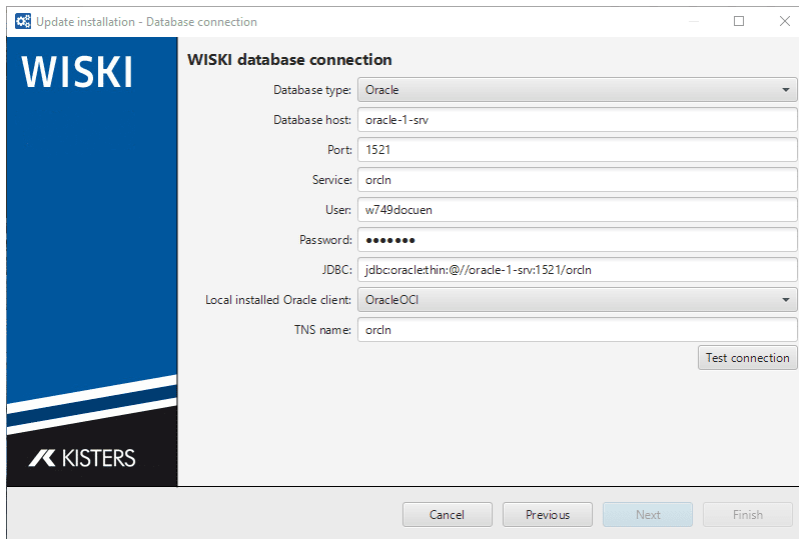
You can modify the server type if necessary and apply different sizes to the configured processes of the server, like **INTERACTIVE**, **AUTOMATION** or **CALCULATION**.

You can configure all nodes of your server individually or use the **SINGLE NODE** option for **INTERACTIVE**, **AUTOMATION**, **KIScript** and **CALCULATION**. This will reduce the number of nodes in your WSM Service Monitor.



In the next step you enter the WISKI server database connection.

After the database connection has successfully been tested, you can activate the **[FINISH]** button.



Update installation - Database connection

WISKI database connection

Database type: Oracle

Database host: oracle-1-srv

Port: 1521

Service: orcln

User: w749docuen

Password:

JDBC: jdbc:oracle:thin:@//oracle-1-srv:1521/orcln

Local installed Oracle client: OracleOCI

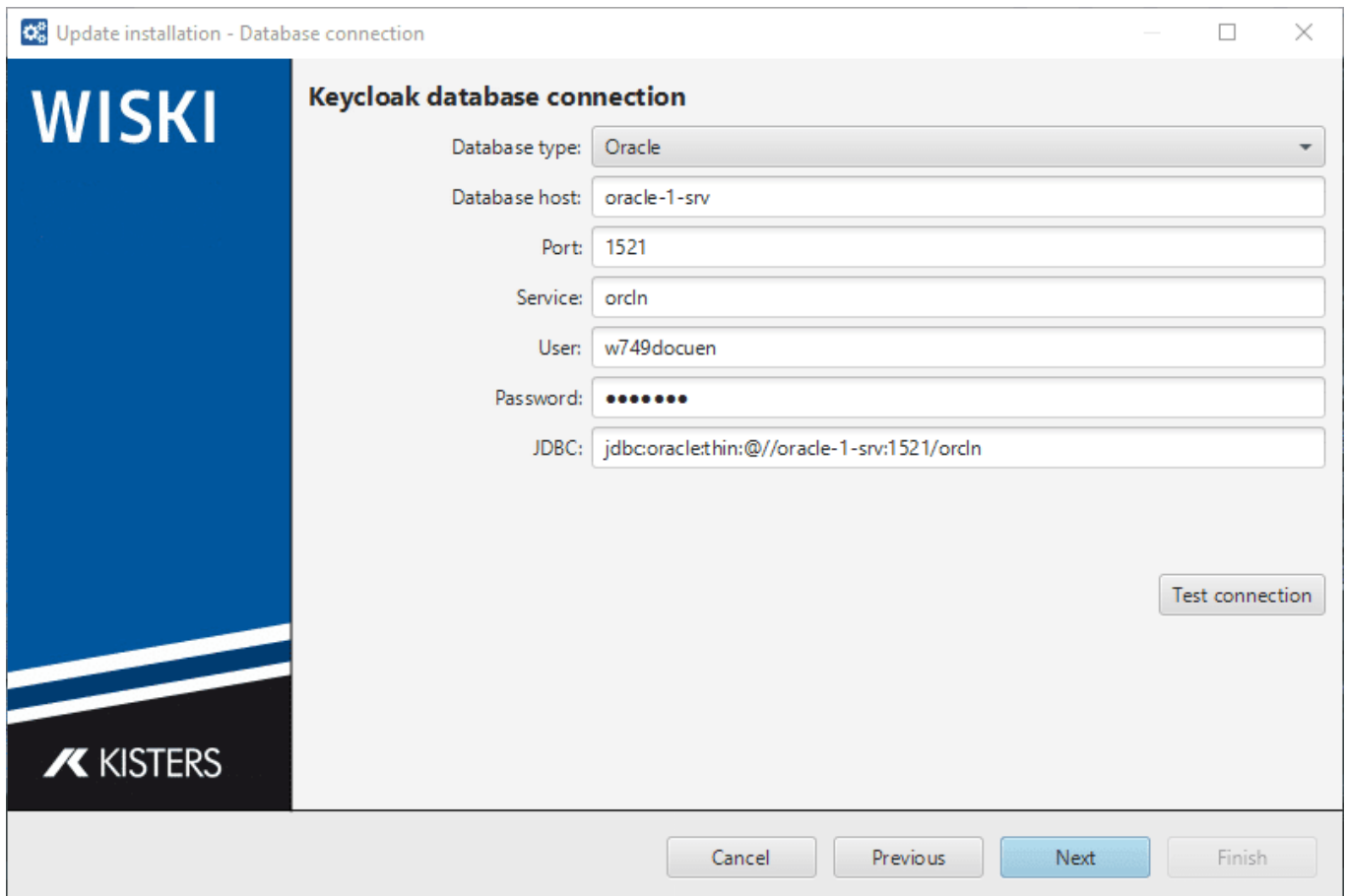
TNS name: orcln

Test connection

Cancel Previous Next Finish

In the next step, enter your credentials for the Keycloak database connection. Before the **[NEXT]** button is activated, the connection must be tested. You can do this by clicking on **[TEST CONNECTION]**.

Note: We recommend using your own database schema for Keycloak. It is possible to install Keycloak inside the WISKI schema, but we do not recommend this for security reasons.



Update installation - Database connection

Keycloak database connection

Database type: Oracle

Database host: oracle-1-srv

Port: 1521

Service: orcln

User: w749docuen

Password:

JDBC: jdbc:oracle:thin:@//oracle-1-srv:1521/orcln

Test connection

Cancel Previous Next Finish

If you have selected the **kidsm** package in the previous package selection, you now have to enter your credentials for the KiDSM database connection. Before the **[NEXT]** button is activated, the connection must be tested. You can do this by clicking on **[TEST CONNECTION]**.

Update installation - Database connection



KiDSM database connection

Database type:

Database host:

Port:

Service:

User:

Password:

JDBC:

If you have selected the **kidsm** package in the previous package selection, you now can define the memory settings of the individual KiDSM modules.

Update installation - KiDSM memory setting

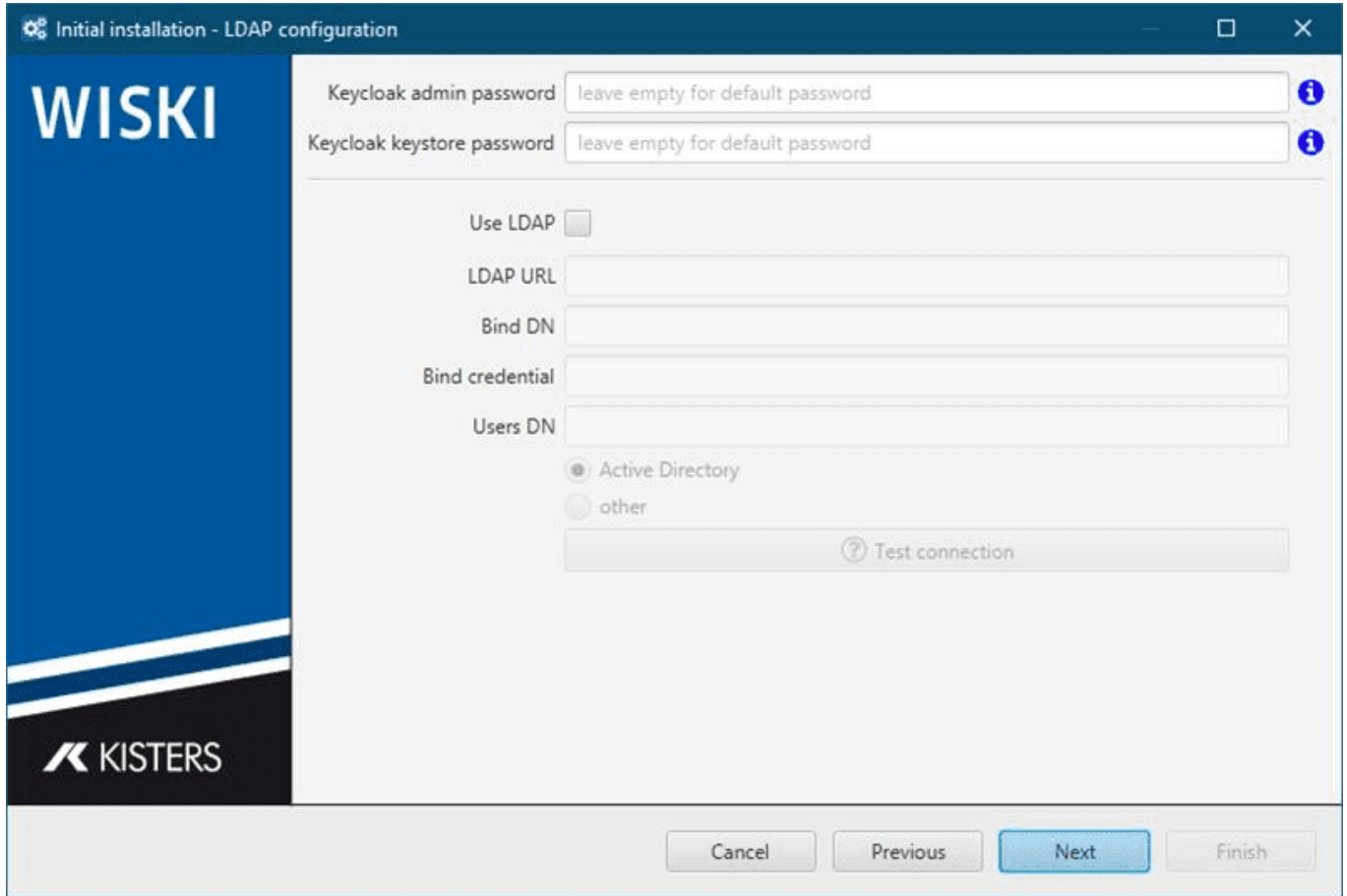


Connector server	<input type="text" value="256"/>	MB
Time scheduler	<input type="text" value="128"/>	MB
Task manager	<input type="text" value="128"/>	MB
Executor	<input type="text" value="128"/>	MB
Message broker	<input type="text" value="128"/>	MB
Transfer tools	<input type="text" value="1"/>	GB
Reverse proxy	<input type="text" value="512"/>	MB

The next window asks for the Keycloak admin password. Note that the Keycloak admin user is named **keycloak** and its default password is **12years**.

Optionally, the configuration of the LDAP backend can be enabled by clicking **USE LDAP**.

Fill in your LDAP credentials.



The screenshot shows a window titled "Initial installation - LDAP configuration". On the left is a blue sidebar with the "WISKI" logo and the "KISTERS" logo at the bottom. The main area contains the following fields and controls:

- Keycloak admin password**: A text field with the placeholder "leave empty for default password" and an information icon.
- Keycloak keystore password**: A text field with the placeholder "leave empty for default password" and an information icon.
- Use LDAP**: A checkbox that is currently unchecked.
- LDAP URL**: A text field.
- Bind DN**: A text field.
- Bind credential**: A text field.
- Users DN**: A text field.
- Two radio buttons: **Active Directory** (selected) and **other**.
- Test connection**: A button with a question mark icon.

At the bottom right, there are four buttons: **Cancel**, **Previous**, **Next** (highlighted in blue), and **Finish**.

More information on this step can be found [here](#).

In the next step, the Dispatcher login can be specified.

Note: This dialog appears only if the default password of the Admin user has been changed.



Update installation

Dispatcher connection

User:

Password:

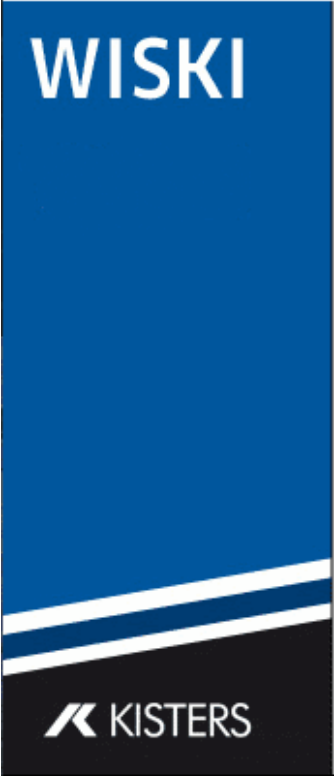
Tenant: ⓘ

Host:

Port:

If you have selected the **portal** package in the previous package selection you now can change the password of the default **PortalAdmin** user.

Note: The default password is preset. Do only modify it if the password has been changed in the WISKI client GUI.



Update installation

Portal administration user

User:

Password: ⓘ

In this dialog you can specify the Windows services to be installed and the user running it.

Update installation - Install WINDOWS services

WISKI

KISTERS

☐ **WISKI Server** ☒ System ☐ Other
 User Password

☐ **KiDSM** ☒ System ☐ Other
 User Password

☐ **KiRollingOptimizer** ☒ System ☐ Other
 User Password

☐ **WISKIDataProvider** ☒ System ☐ Other
 User Password

Cancel Previous **Next** Finish

If you have selected the **kiwis** package in the previous package selection you now can customise the KiWIS configuration. In the lower part of the window you now have to enter your credentials for the KiWIS dispatcher settings.

Update installation - KiWIS configuration

WISKI

KISTERS

KiWIS configuration

Cache settings ☒ Default cache settings
☐ Live data
☐ Fast response times

Timezone GMT

WDP Server vm-lavie-w7prod.kisters.de

KiWIS dispatcher settings

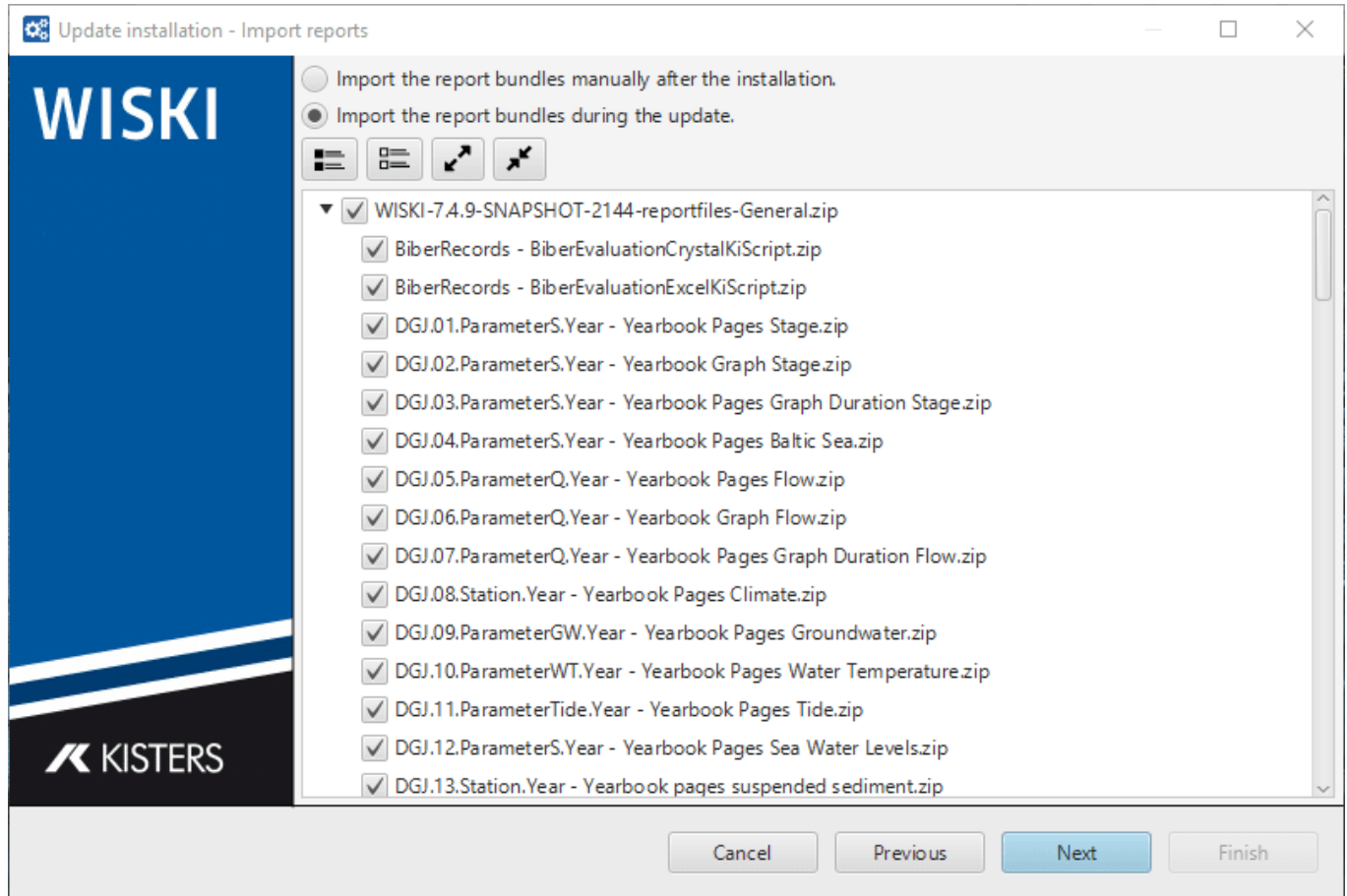
Host vm-lavie-w7prod.kisters.de

User Admin

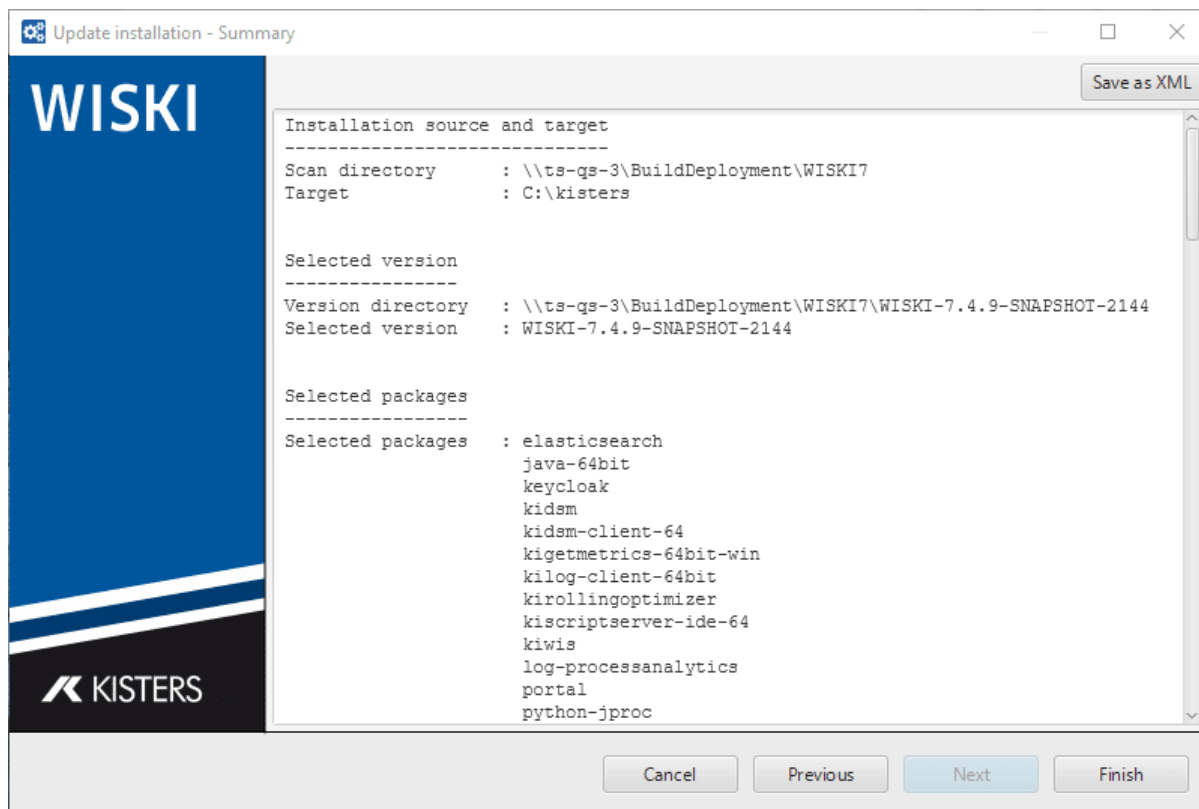
Password

Cancel Previous **Next** Finish

The next step is to select the report bundles which are to be imported. Alternatively you can import the report bundles after the installation.

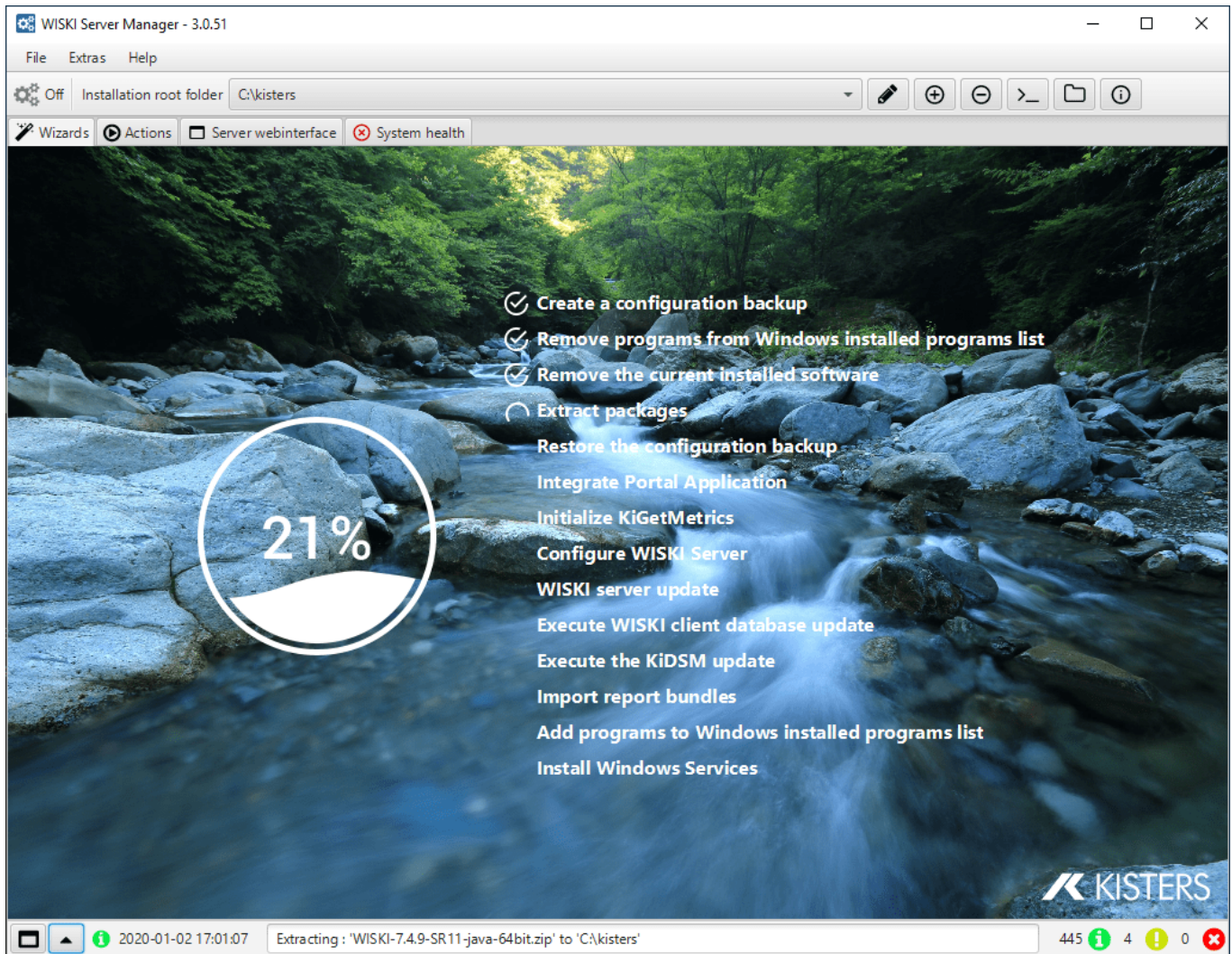


The last dialog shows you a summary of the update configuration.



Click **[FINISH]** to start the update process.

A graphical progress bar will show you the progress of the update.



After the update, the WINDOWS services have to be re-installed. See ch. [Windows services](#) for details.

9 Duplicating KISTERS WISKI7 Server Database

Beside the two scenarios of setting up a initial installation and updating a system, there is a third very common use case. The WISKI database is often duplicated, e.g. to create a test system out of the productive system. The WISKI database can contain all together three database schemas:

- the WISKI7 schema,
- the WISKI server dispatcher schema and
- the WISKI server tenant database schema.

This can be done manually, but it is recommended to use the WISKI Server Manager.

Warning: The connection entries from the WISKI server dispatcher schema to the WISKI7 schema and the connection information from the WISKI server dispatcher to the tenant database schema are stored inside the dispatcher schema. Therefore it is important that you follow the next steps very carefully, otherwise the original system can be damaged!

The steps described hereafter are only valid for the following conditions:

- WISKI7 / WISKI server schemas are all in one schema.
- For your new (duplicated) system you already have an installed WISKI7 / WISKI server software, otherwise read these chapters before starting the actual duplication:
 - Extracting the software packages
 - Installing Perl

Note: If you want to set up a test system which uses a newer software version than the original system you have first to install the same version like the system you copied the database from! The next step is to update the copied database!

Duplicating databases comprises these working steps:

- [Exporting Original Database](#)^[89]
- [Configuring Connection Settings to WISKI7 Server Database Manually](#)^[92]
- [Updating Dispatcher Database Connections](#)^[94]
- [Updating the Client Schema](#)^[99]

9.1 Exporting Original Database

To export the original database to be used for duplicating:

- Shut down all import services.
- Make sure all clients are disconnected.
- Open the KiTSM web interface under <http://localhost:7420/TsmSAdmin/logon.do>.
- Navigate to **TENANTS > TENANTS**, select your desired tenant (e.g. **WISKI**); you will be guided to the **TENANT ADMINISTRATION: WISKI** tab.
- Click on the link in the **STATE** column for the database storage with the name **tsmDb**.
- The link should be green and be named **online**.

Tenant		Tenant in shared db		MaiDomain	
Id	Tenant	State	Session duration	Template	
1001	MaiDomain	online	300		
2001	tcaExDb	online	yes	JDBC URL	Db user
4001	tsmDb	online	no	JDBC URL	Db user
5001	kiiosys	available	yes	Version up to date	Version
6001	tsmsrver	available	yes	Version up to date	Version

- Change the state to **admin** and click the **[SAVE]** button.

Change tenant state

Tenant: MaiDomain

Tenant database storage: tsmDb

Tenant state: admin

restricted

online

Save

- Now the link is red and is named **admin**.

MaiDomain									
Id		Tenant	State	Session duration		Template			
1001		MaiDomain	online	300					
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
2001		tcaExDb	online	yes	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72			
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
3001		tsmDb	admin	no	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72	tsmserver, tsm-importserver, tsm-queue-server, kiiosys		
Id		Application-db	State	Exists	Version up to date	Version	Data up to date		
3001		kiiosys	available	yes	yes	2_06	yes		
4001		tsmserver	available	yes	yes	10_03	yes	Tsm object history false Data set type wiski	

- Click the link again and change the state of the database storage to **configured**; confirm with [SAVE].

MaiDomain									
Id		Tenant	State	Session duration		Template			
1001		MaiDomain	online	300					
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
2001		tcaExDb	online	yes	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72			
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
3001		tsmDb	configured	no	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72	tsmserver, tsm-importserver, tsm-queue-server, kiiosys		
Id		Application-db	State	Exists	Version up to date	Version	Data up to date		
3001		kiiosys	available		no		no		
4001		tsmserver	available		no		no	Tsm object history false Data set type wiski	

- Click on the  button that appears on the right side. The tool tip should read **Unregister database storage**.

MaiDomain									
Id		Tenant	State	Session duration		Template			
1001		MaiDomain	online	300					
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
2001		tcaExDb	online	yes	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72			
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
3001		tsmDb	configured	no	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72	tsmserver, tsm-importserver, tsm-queue-server, kiiosys		
Id		Application-db	State	Exists	Version up to date	Version	Data up to date		
3001		kiiosys	available		no		no		
4001		tsmserver	available		no		no	Tsm object history false Data set type wiski	

- On the next page click the red [YES, UNREGISTER] button. You will return to the **TENANT ADMINISTRATION: WISKI** tab; the **tsmDb** database entry will be gone.

MaiDomain									
Id		Tenant	State	Session duration		Template			
1001		MaiDomain	online	300					
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
2001		tcaExDb	online	yes	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72			

- Shut down the JProcController service. There are various ways to do that.
 - Use the WISKI Server Manager, navigate to the tab **ACTIONS** and execute the command **Service - stop JProcController**.
 - Or shut down the WINDOWS service (if installed).
 - Or navigate to the `<root>\wiski\bin` directory and execute the batch file `jprocontroller_stop.bat`.
- Wait until all services have been shut down.
- To check if the system was stopped completely, start the WSM Service Monitor and check the status.
 - Use the WISKI Server Manager, navigate to the tab **ACTIONS** and execute the command **Client - JProcController Console**.
- Export the database user (see your ORACLE or Microsoft SQL Server documentation for details).
- After the export, start the original server again.
- Open the KiTSM web interface again.
- Navigate to **TENANTS > TENANTS**, select your desired tenant (e.g. **WISKI**) and you will be guided to the **TENANT ADMINISTRATION: WISKI** tab.
- Click the button  in the left dialog part where the tool tip text says Register existing database storage.

Tenant: Tenant in shared db

Create database storage

MaiDomain

Database storage key:

Database vendor: ms-sql2005

JDBC URL:

Database user:

Database user password:

Repetition of database user password:

Database-Pool minimum connections: 2

Database-Pool shrink threshold: 8

Database-Pool shrink start time [milliseconds]: 1200000

Database-Pool shrink time factor: 3

The action may last few minutes.

- Enter the database key **tsmDb**.
- Enter all information for the database connection.
- You can use the `dcsmgr-dispatcher-custom.properties` file to make life easier.
- Click the **[CHECK STORAGE]** button.
- On the next page click the **[REGISTER AND SAVE]** button.

Tenant: Tenant in shared db

Create database storage

Result of check

Tenant name in origin: MaiDomain

Cluster identifier of origin (should be empty):

Identifier of this cluster: QV78Ve1G2SF304axULE+UA==

Storage identifier of origin (should be empty):

Cluster URL of origin (should be empty):

Dispatcher last check date:

Tadm Schema exists: true

Application-db	State	Exists	Version up to date	Version	Data up to date
kiosys	created	yes	yes	2_06	yes
tsmserver	created	yes	yes	10_03	yes

Tsm object false history

Data set type wiski

MaiDomain

Database storage key: tsmDb

Database vendor: oracle

JDBC URL: jdbc:oracle:thin:@oracle-10g-srv:1521:qswiski

Database user: w7joerg72

Database user password: *****

Repetition of database user password: *****

Database-Pool minimum connections: 2

Database-Pool shrink threshold: 8

Database-Pool shrink start time [milliseconds]: 1200000

Database-Pool shrink time factor: 3

The action may last few minutes.

- Click on the link in the **STATE** column for the database storage with the name **tsmDb** entry which says **configured**.

Tenant: Tenant in shared db

Id		Tenant	State	Session duration	Template
1001		MaiDomain	online	300	

Id	Database storage	State	Extra db	JDBC URL	Db user	Applications
2001	tcaExDb	online	yes	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiski	w7joerg72	
3001	tsmDb	configured	no	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiski	w7joerg72	tsmserver, tsm-importserver, tsm-queue-server, kiosys

Id	Application-db	State	Exists	Version up to date	Version	Data up to date
3001	kiosys	available		no		no
4001	tsmserver	available		no		no

Tsm object history false

Data set type wiski

- Change the state to **admin** and click the **[SAVE]** button.

Application Db plugins | Groups | Tenants | KiTSM tenant: MaiDomain | Tenant administration: MaiDomain

Change tenant state

Tenant: MaiDomain

Tenant database storage: tsmDb

Tenant state: admin
restricted
online

- Now the link should read **admin**.

MaiDomain									
Id		Tenant		State	Session duration			Template	
1001		MaiDomain		online	300				
Id		Database storage	State	Extra db	JDBC URL		Db user	Applications	
2001		tcaExDb	online	yes	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki		w7joerg72		
Id		Database storage	State	Extra db	JDBC URL		Db user	Applications	
3001		tsmDb	admin	no	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki		w7joerg72	tsmserver, tsm-importserver, tsm-queue-server, kiiosys	
Id		Application-db	State	Exists	Version up to date	Version	Data up to date		
3001		kiiosys	available	yes	yes	2_06	yes		
4001		tsmserver	available	yes	yes	10_03	yes		
		Tsm object history/false							
		Data set type/wiski							

- Change the state to **online** and click the **[SAVE]** button.

Home

TSM Cluster

Log Viewer

Tenants

Authorisation

Preferences

Application Db Plugins

Groups

Tenants

1 - 1 / 1

Id	Tenant	State	Session Duration	Groups	Template	Auth. Type Key	
Id	Database storage	State	Extra db	JDBC URL	Db user		
	Id	Application-db	State	Exists	Version up to date	Version	Data up to date
1001	MaiDomain	online	900				
1001	tsmDb	online	no	jdbc:sqlserver://kn-jheimbuchner\SQLEXPRESS:1433;DatabaseName=wiski	wiskiadm		
1001	tsmserver	available	yes	yes	10_11	yes	
2001	kiiosys	available	yes	yes	2_09	yes	
2001	tcaExDb	online	yes	jdbc:sqlserver://kn-jheimbuchner\SQLEXPRESS:1433;DatabaseName=wiski	wiskiadm		

- Import the database in a new schema (see your ORACLE or MSSQL Server documentation for details).

9.2 Configuring Connection Settings to WISKI7 Server Database Manually

The WISKI server connects through the JDBC interface straight to the database (MSSQL and ORACLE). JDBC stands for JAVA DATABASE CONNECTIVITY and does not need any database client installation on the KiTSM Server (ORACLE CLIENT, ODBC driver, etc).

Only if WISKI7 is installed on the same machine as the KiTSM Server, a database client installation is required. During the initial installation the KiTSM Dispatcher database connection is configured by an initialising Perl script. If you want to set up a KiTSM working with a duplicated database you have to do this manually.

Note: In the case that the database has been copied, it is also possible to copy the file from the original system and modify.

- Change to `<root>\wiski\services\dcserver\server_tsmdispatcher\configuration`.
- Open the `dcsmgr-dispatcher-default.properties` file into an ASCII Editor.
- Apply the JDBC Configuration settings as described below.
- Save the changes as `dcsmgr-dispatcher-custom.properties`.

```

tsm-dispatcher.properties - Notepad
File Edit Format View Help
#- dispatcher properties
tsm-dispatcher.log4j.initfile=tsm-dispatcher-log4j.properties
#-TSM-System Database Connection URL
#- Oracle
#- Oracle standard port = 1521
#- Configuration example for Oracle JDBC Thin Connections
#- tsms.dispatcher.database.url=jdbc:oracle:thin:@<hostname>:<port>:<SID>
#- Configuration example for Oracle Multi-Threading Services JDBC Thin Connections
tsms.dispatcher.database.url=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(P
#- Informix
#- tsms.dispatcher.database.url=jdbc:informix-sqli://<hostname>:<port>/<database>:INF
#- Informix standard port = 1526
#- MS SQL Server 2000
#- tsms.dispatcher.database.url=jdbc:microsoft:sqlserver://mssql-srv-01:1433;DatabaseName
#- SQL Server standard port = 1433
#- MS SQL Server 2005
#- tsms.dispatcher.database.url=jdbc:sqlserver://mssql-srv-02:1433;DatabaseName=wiski7_
#- SQL Server standard port = 1433
tsms.dispatcher.database.user=w7demo
# generated encrypted password by tsm-dispatcher-script.
# example: ./tsm-dispatcher.sh admin createpassword 123abc
# result: crypt_password[1] ewrPMB1Cody=
tsms.dispatcher.database.passwd.encrypted=Fvo1vIR/Afo=
#####
#- database storage parameter:
#- 3 different schema:
#-   tdis - dispatcher, tsm - tsm domain, tsc - tsm domain time series container
#-
#- system type (default normal, currently supported: normal)
#tsms.dispatcher.database.storage.tdis.systemtype=normal
#tsms.dispatcher.database.storage.tsm.systemtype=normal
#tsms.dispatcher.database.storage.tsc.systemtype=normal
#-
#- kind of physical storage clause to use: (default is false)
#- use individual storage clauses (default is false, currently no configuration avail
#tsms.dispatcher.database.storage.tdis.useclause=true
#tsms.dispatcher.database.storage.tsm.useclause=true
#tsms.dispatcher.database.storage.tsc.useclause=true
#-
#- use table spaces for all objects (default false) - requires tablespaces and their
#tsms.dispatcher.database.storage.tdis.usetablespace=true
#tsms.dispatcher.database.storage.tsm.usetablespace=true
#tsms.dispatcher.database.storage.tsc.usetablespace=true
#-
#- if usetablespace=true a tablespaces assignment for dbtype is needed.

```

① JDBC Database setting

② User name

③ Password

This configuration is explained in detailed in the following sub-chapters:

- [Connection Entries for ORACLE](#)^[93]
- [Connection Entries for Microsoft SQL Server](#)^[93]
- [Set User Name and Password](#)^[93]

9.2.1 Connection Entries for ORACLE

- Enter this line in the `dcsmgr-dispatcher-custom.properties` file:

```
tsms.dispatcher.database.url=jdbc:ORACLE:thin:@$MY_ORACLE_SERVER:1521:$MY_SID
```

- Replace `$MY_ORACLE_SERVER` with the machine name or with the IP address of the ORACLE database server.
- Replace `$MY_SID` with the value of the SID (ORACLE System Identifier of the database, usually **ORCL**) of the database.

9.2.2 Connection Entries for Microsoft SQL Server

- Enter this line in the `dcsmgr-dispatcher-custom.properties` file:

```
tsms.dispatcher.database.url=jdbc:sqlserver://$MY_SQL_SERVER:1433;DatabaseName=$WISKI_DATABASE
```

- Replace `$MY_SQL_SERVER` with the machine name of the MSSQL database server.
- `$WISKI_DATABASE` must be replaced by the name of the WISKI7/KITSM database.

9.2.3 Set User Name and Password

User name and password needs to be configured, use the section of the `dcsmgr-dispatcher-custom.properties` property file.

```
tsms.dispatcher.database.user=wiski7
# generated encrypted password by tsm-dispatcher-script:
# example: ./tsm-dispatcher.sh admin createpassword 123abc
```

```
# result: crypt password rWrpMBlCodY=
tsms.dispatcher.database.passwd.encrypted=FVoiVIR/Afo=
```

All respective terms must be replaced by your desired user name and password appropriately.

Replace **wiski7** with the name of the database user created earlier.

The password **FVoiVIR/Afo=** means **12years** if the WISKI7 database has got a different password you need to generate an encrypted code for the password. You can do this using the KiTSM batch file **tsm-dispatcher.pl** from the directory **\tsm\bin**.

Command: **<root>\wiski\services\tsm\bin>tsm-dispatcher.pl createpassword <password>**

Now the dispatcher is able to connect to the dispatcher schema. But inside of the dispatcher schema are connection entries to the other parts of the system which must be reconfigured, therefore you have to start the system and access the KiTSM Web interface.

9.3 Updating Dispatcher Database Connections

After successfully configuring the **DCSMGR-DISPATCHER-CUSTOM.PROPERTIES** file the system must be configured and started. The next step will create a configuration and migrate the dispatcher database.

Navigate to **<root>\wiski\services\wiski-server\bin** and execute **updatew7.pl 0-7**.

Note: The tenant and user/password can be specified in the command line when executing **updatew7.pl**:

updatew7.pl -domain=<tenant> -user=<user> -password=<pwd>

Info: All available steps can be displayed using the command: **updatew7.pl -s**

After successfully running the **updatew7.pl** script the Dispatcher should be running. Now the procedure continues in the server web interface.

There are two ways to open the web interface:

- Using the WISKI Server Manager, navigate to the tab **SERVER WEBINTERFACE**. Click the button **[CONNECT]**, this should log you in automatically.
- The second way to log in is to run the batch file which uses the default system web browser: Navigate to **<root>\wiski\bin** and execute **kitsm_webinterface.bat**.

If a login screen appears, use the following login data:

Setting	Value
Tenant	KiAdminDomain
Role	KiClusterAdmin
User	KiClusterAdmin
Password	your password, the default password is 123abc

After opening the web interface you will see the welcome screen. The system has already noticed that it was copied. The tenants have been closed automatically. This means that no KiTSM service can access the database. This is a protective mechanism of the KiTSM to protect your database.

Now these steps have to be followed:

- [Checking and Correcting the TCA Database Storage](#)⁹⁵
- [Registering the KiTSM Tenant Database Storage](#)⁹⁶
- [Running the Complete Update Process](#)⁹⁷
- [Checking the Background Process Configuration](#)⁹⁸
- [Registering the New Dispatcher Database URL](#)⁹⁹

9.3.1 Checking and Correcting the TCA Database Storage

Note: In the case that the storages have not been unregistered before export you can not set the status to **online**. Execute the following SQL statements to remove the registration from the database.

```
delete from systemparam_tadm where spatadm_key='ClusterIdentifier';
delete from systemparam_tadm where
spatadm_key='ClusterStorageIdentifier';
delete from systemparam_tadm where
spatadm_key='ClusterVerificationURL';
commit;
```

Now the status of the storages can first be set to **admin** and then to **online**.

- Navigate to **TENANTS > TENANTS**, select your tenant (e.g. **WISKI**) and you will be guided to the **TENANT ADMINISTRATION: WISKI** tab.

Tenant				
Tenant in shared db				
MaiDomain				
Id	Tenant	State	Session duration	Template
1001	MaiDomain	online	300	

- Click the link in the **STATE** column for the **tcaExDB** storage; the link should be green and named **online**.
- Change the state from **online** to **admin** and click the **[SAVE]** button.
- Now the link is red and is named **admin**.

Tenant				
Tenant in shared db				
MaiDomain				
Id	Tenant	State	Session duration	Template
1001	MaiDomain	admin	300	

- Click the link again and change the state of the **tcaExDB** storage to **configured**, confirm with the **[SAVE]** button.

Tenant				
Tenant in shared db				
MaiDomain				
Id	Tenant	State	Session duration	Template
1001	MaiDomain	configured	300	

- Click the  button to change the storage configuration:

JDBC URL	<JDBC connection string>
Database user	<The new database user>
Database password	<Password of the database user>

- Save the new settings.

Change database storage

MaiDomain tcaExDb

Database storage key: tcaExDb

Database vendor: oracle

JDBC URL:

Database user:

Database user password:

Repetition of database user password:

Database-Pool minimum connections: 2

Database-Pool shrink threshold: 8

Database-Pool shrink start time [milliseconds]: 1200000

Database-Pool shrink time factor: 3

Save

9.3.2 Registering the KiTSM Tenant Database Storage

- Navigate to **TENANTS > TENANTS**, select your tenant (e.g. **WISKI**) and you will be guided to the Tenant administration: **WISKI** tab.

Click the button  in the left dialog part where the tool tip says Register existing database storage.

Tenant Tenant in shared db

Create database storage

MaiDomain

Database storage key:

Database vendor: ms-sql2005

JDBC URL:

Database user:

Database user password:

Repetition of database user password:

Database-Pool minimum connections: 2

Database-Pool shrink threshold: 8

Database-Pool shrink start time [milliseconds]: 1200000

Database-Pool shrink time factor: 3

The action may last few minutes. Check storage

- Enter the database key **tsmDb**.
- Enter all information for the database connection. You can use the `tsm-dispatcher-custom.properties` file to make life easier.
- Click the **[CHECK STORAGE]** button.
- On the next page select **[REGISTER AND SAVE]**.

Tenant Tenant in shared db

Create database storage

Result of check

Tenant name in origin: MaiDomain

Cluster identifier of origin (should be empty):

Identifier of this cluster: QV78Ve1G2SF304axULE+UA==

Storage identifier of origin (should be empty):

Cluster URL of origin (should be empty):

Dispatcher last check date:

Tadm Schema exists: true

Application-db	State	Exists	Version up to date	Version	Data up to date	
kiiosys	created	yes	yes	2_06	yes	
tsmserver	created	yes	yes	10_03	yes	

Tsm object false history

Data set type wiski

MaiDomain

Database storage key: tsmDb

Database vendor: oracle

JDBC URL: jdbc:oracle:thin:@oracle-10g-srv:1521:qswiski

Database user: w7joerg72

Database user password: *****

Repetition of database user password: *****

Database-Pool minimum connections: 2

Database-Pool shrink threshold: 8

Database-Pool shrink start time [milliseconds]: 1200000

Database-Pool shrink time factor: 3

The action may last few minutes. Register and save

- Click the link in the State column for the **tsmDb** storage; the link should be red and named configured.

Tenant: Tenant in shared db									
MaiDomain									
Id		Tenant	State	Session duration			Template		
1001		MaiDomain	online	300					
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
2001		tcaExDb	online	yes	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72			
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
3001		tsmDb	configured	no	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72	tsmserver, tsm-importserver, tsm-queue-server, kiosys		
Id		Application-db	State	Exists	Version up to date	Version	Data up to date		
3001		kiosys	available		no		no		
4001		tsmserver	available		no		no	Tsm object history false	
								Data set type wiski	

- Change the state to admin and click the [SAVE] button.

Application Db plugins	Groups	Tenants	KITSM tenant: MaiDomain	Tenant administration: MaiDomain
------------------------	--------	---------	-------------------------	----------------------------------

Change tenant state

Tenant: MaiDomain

Tenant database storage: tsmDb

Tenant state:

admin
restricted
online

Save

- Click the link again and change the state of the **tsmDb** storage to admin, confirm with the [SAVE] button.

MaiDomain									
Id		Tenant	State	Session duration			Template		
1001		MaiDomain	online	300					
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
2001		tcaExDb	online	yes	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72			
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
3001		tsmDb	admin	no	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72	tsmserver, tsm-importserver, tsm-queue-server, kiosys		
Id		Application-db	State	Exists	Version up to date	Version	Data up to date		
3001		kiosys	available	yes	yes	2_06	yes		
4001		tsmserver	available	yes	yes	10_03	yes	Tsm object history false	
								Data set type wiski	

- Change the state to online and click the [SAVE] button.

Tenant: Tenant in shared db									
MaiDomain									
Id		Tenant	State	Session duration			Template		
1001		MaiDomain	online	300					
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
2001		tcaExDb	online	yes	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72			
Id		Database storage	State	Extra db	JDBC URL	Db user	Applications		
4001		tsmDb	online	no	jdbc:oracle:thin:@oracle-10g-srv:1521:qswiki	w7joerg72	tsmserver, tsm-importserver, tsm-queue-server, kiosys		
Id		Application-db	State	Exists	Version up to date	Version	Data up to date		
5001		kiosys	available	yes	yes	2_06	yes		
6001		tsmserver	available	yes	yes	10_03	yes	Tsm object history false	
								Data set type wiski	

- Save the new setting.

Note: You can use the `dcsmgr-dispatcher-custom.properties` file to make life easier.

9.3.3 Running the Complete Update Process

To configure the cluster, the easiest way is to run the `updatew7.pl` script again. When running the script the user name and password of the WISKI user has to be specified in order to execute the last steps.

```
updatew7.pl -domain=<tenant> -user=<user> -password=<pwd>
```

An easier way to execute the script is a wrapper batch file.

Navigate to `<root>\wiski\bin` and execute the batch file `wiski-server_update_db.bat`.

9.3.4 Checking the Background Process Configuration

- | KitSMM tenant | Background processes | Background calculation | History migration | Tenant properties | | | | | |
|---------------|----------------------|--|-------------------|-------------------|--|--|--|--|--|
| | | <div> <div>+</div> <div>Key</div> <div>Type</div> <div>active TSM-Server</div> <div>Server-Id</div> <div>Number</div> <div>Threads</div> <div>Min. memory</div> <div></div> </div> | | | | | | | |
| | | <div> <div>+</div> <div>7042</div> <div>calculation trigger marking</div> <div>yes</div> <div>7001</div> <div>1</div> <div></div> <div>150</div> <div></div> </div> | | | | | | | |
| | | Instances | | | | | | | |
| | | <div> <div></div> <div>Key</div> <div>State</div> <div>Session</div> <div>update time</div> <div>created</div> <div>Usage</div> <div></div> </div> | | | | | | | |
| | | <div> <div>+</div> <div>8042</div> <div>Execution of pending calculation</div> <div>yes</div> <div>7001</div> <div>1</div> <div></div> <div>150</div> <div></div> </div> | | | | | | | |
| | | Instances | | | | | | | |
| | | <div> <div></div> <div>Key</div> <div>State</div> <div>Session</div> <div>update time</div> <div>created</div> <div>Usage</div> <div></div> </div> | | | | | | | |

KITSM tenant

Background processes

Background calculation

History migration

Tenant properties

Change Background Processes

Tenant: MaiDomain

Type: calculation trigger marking

Next

- KitSMM tenant

Background processes

Background calculation

History migration

Tenant properties

Change Background Processes

Tenant: MaidDomain

TSM-Server: 192.168.54.225#1 16001

Type: calculation trigger marking:

Active: yes

Number: 2

Minimum free memory [Mb]: 150

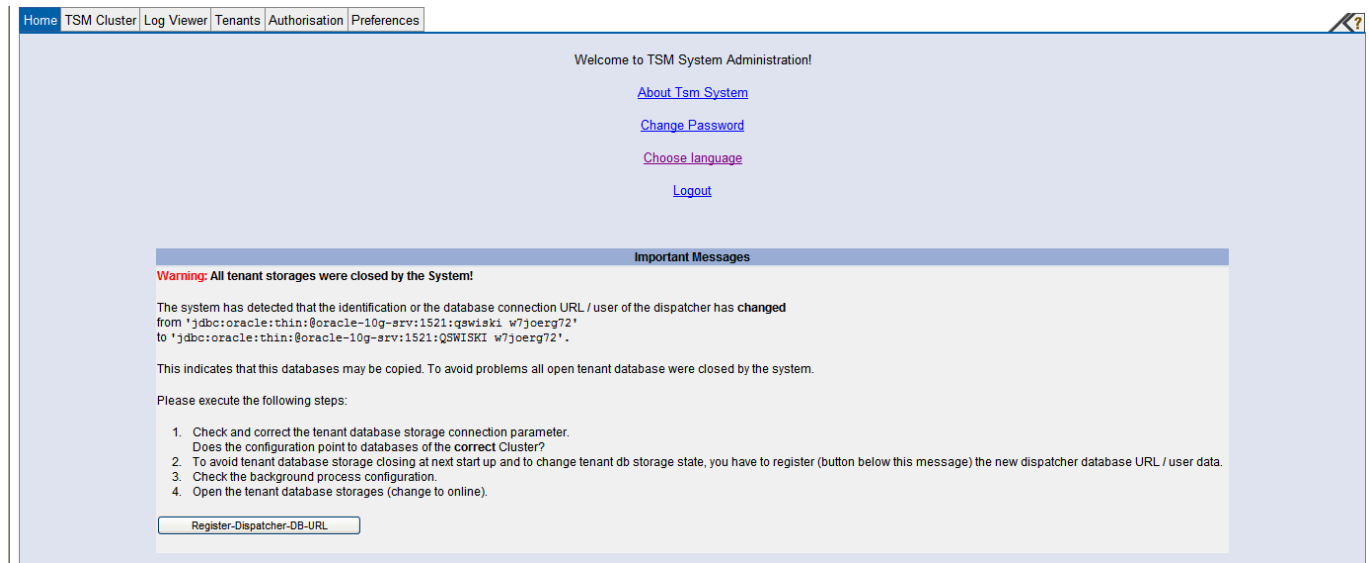
Save

Reset

Note that you have to create each type of background process, so you end up with two background processes in total.

9.3.5 Registering the New Dispatcher Database URL

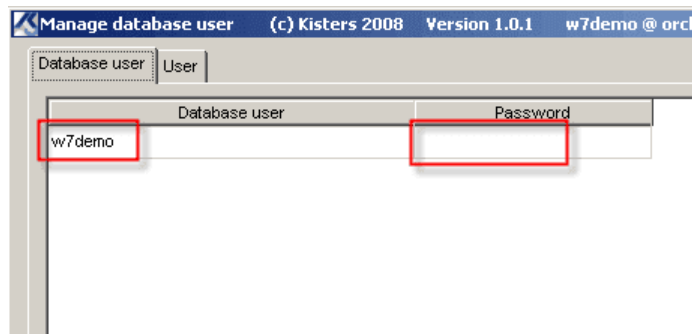
The last step is to navigate to the main page of the web interface and click the **[REGISTER DISPATCHER DB URL]** button below the message informing you that all tenant databases had been closed by the system.



9.4 Updating the Client Schema

The last step of the duplicating procedure is that you start WISKI7 with the **/DBadmin** option (part of the installation) to correct the database user and password.

Fill in the login dialog with the connection details of your system, then change the database user and password that was stored in the existing schema.



Now you have a fully operational copy of your WISKI7/KiTSM system.

The next steps could be updating the system to a newer software version.

10 WISKI Server Initial Installation (Manual Installation without WSM)

The manual installation of a complete WISKI system is divided into the following main areas of functionality:

- [Manual Server Installation Example](#)^[100]
- [WISKI7 Client Installation Tasks](#)^[104]
- [Post-installation Steps](#)^[112]

10.1 Manual Server Installation Example

The KiTSM connects through the JDBC interface straight to the database (MSSQL and ORACLE). JDBC stands for JAVA DATABASE CONNECTIVITY and does not need any database client installation on the Server (ORACLE CLIENT, ODBC driver, etc). Only if WISKI7 is installed on the same machine as the KiTSM Server, a database client installation is required. The setup and configuration of the KiTSM system is mainly operating on ZIP file extraction and running Perl scripts which configure property files and manage the system. The underlying folder structure should not be changed.

In general, you must perform these steps:

- [Put together the packages which should be installed and extract the packages](#)^[100]
- [Initialize the Server Database Part](#)^[100]
- [Install the WISKI Services as a WINDOWS Service](#)^[101]
- [Checking the KiTSM state](#)^[101]

10.1.1 Put together the packages which should be installed and extract the packages

Put together a [package configuration](#)^[10], e.g. if you would like to install a 64-bit WISKI server installation, you need the following packages:

- WISKI-<vn>-<bn>-java-64bit.zip
- WISKI-<vn>-<bn>-kidsm.zip
- WISKI-<vn>-<bn>-kidsm-client-64.zip
- WISKI-<vn>-<bn>-kilog-client-64bit.zip
- WISKI-<vn>-<bn>-services.zip
- WISKI-<vn>-<bn>-wiski-client.zip
- WISKI-<vn>-<bn>-wiski-client-help-all.zip

Create a root installation folder. The KISTERS WISKI installation path on windows platform may not contain any white space characters. A valid installation path could be `c:\kisters`. Copy all archives to that directory. Extract all archives with the option **EXTRACT HERE**.

Note: This manual uses the placeholder `<root>` for the installation folder.

Warning: The Installation path should not have any white space or special characters. Valid path strings contain only a-z, AZ, 0-9.

10.1.2 Initialize the Server Database Part

Navigate with the WINDOWS explorer to `<root>\wiski\bin`.

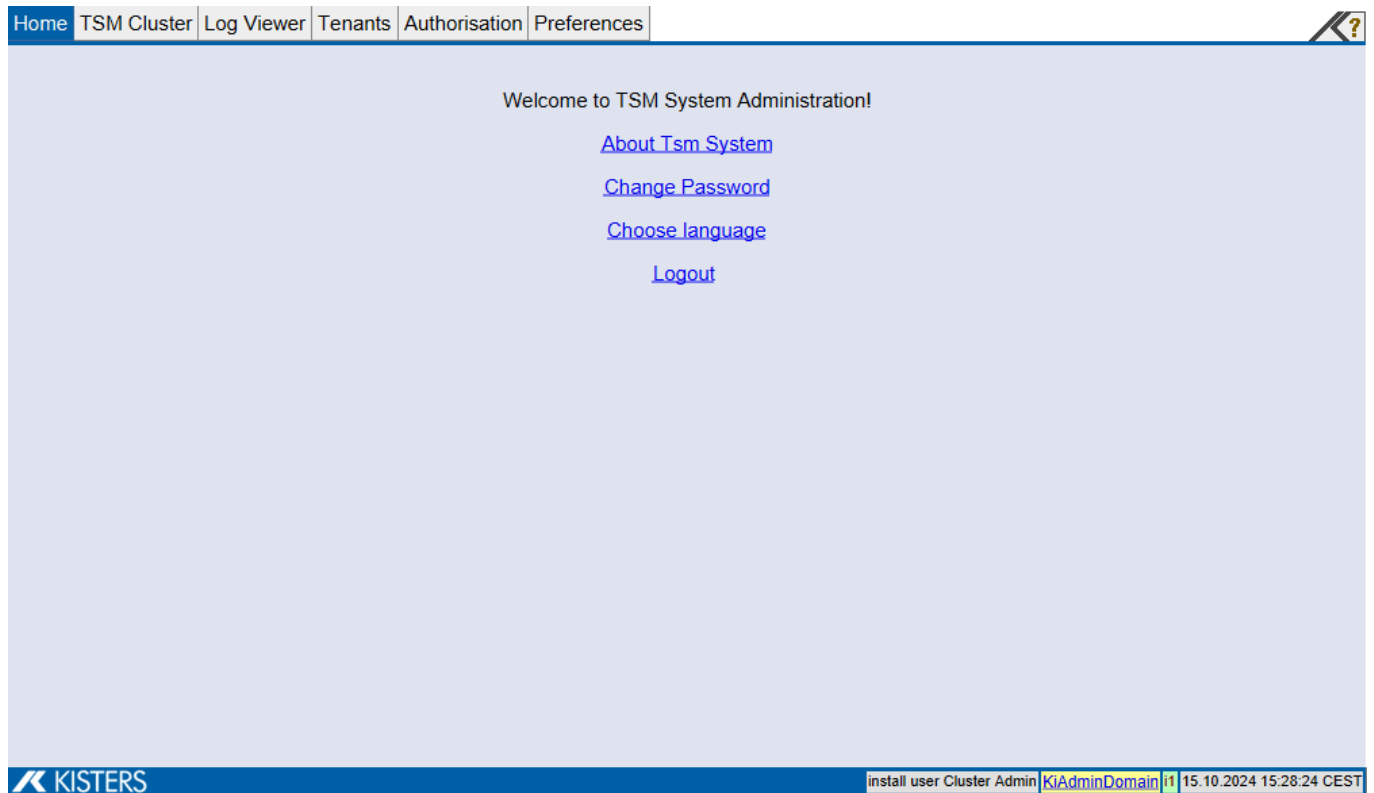
Execute `wiski-server_initialize_db.bat`, fill out the connection data.

10.1.3 Install the WISKI Services as a WINDOWS Service

10.1.4 Checking the KiTSM state

Navigate to `<root>\wiski\bin` and execute `kitsm_webinterface.bat`. This batch file will open the system default browser and log in to the KiTSM Webinterface. For the login the default user name and password are used.

Tenant	KiAdminDomain
Role	KiClusterAdmin
Login	KiClusterAdmin
Password	123abc



Detailed information can be found in the next subchapter(s):

- [Monitoring General KiTSM Server State](#) ¹⁰²

10.1.4.1 Monitoring General KiTSM Server State

A KiTSM Server has two states: a **TSM STATE** and an **ADM STATE**. The TSM state is responsible for the green **RUN** field in the WSM Service Monitor. It indicates that the server process is physically running. To use the running KiTSM server with WISKI7 the **ADM STATE** must be set to **ON**. Normally this is the case, however it is worth to verify this. This check is also necessary, if WISKI7 returns the error message **NO FREE TSM** while logging in to KiTSM.

TSM CLUSTER > CLUSTER NODES

Home

TSM Cluster

Log Viewer

Tenants

Authorisation

Preferences

Controller

Cluster Nodes

Cluster Load

Dispatcher

Cluster Sessions

Background Processes

1 - 5 / 5

Key	Instance Id	Host	Instance Name	Service Name	State	Adm State	Heartbeat Date	Tolerance [min]	Load	URL	Software
1001	105001	kn-jheimbuchner.kisters.de	10	queueserver	on	on	13.11.2013 13:02:29 CET	2	0.00	rmi://kn-jheimbuchner.kisters.de:1099/ClusterNodeControl10	Release_TSM_3-40-7 31.10.2013 14:47:54 CET
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
1001	tsm-queue-server	tsm-queue-server	7442	on	on	0 (0)	0		true		
2001	104001	kn-jheimbuchner.kisters.de	20	automation	on	on	13.11.2013 13:02:29 CET	2	0.00	rmi://kn-jheimbuchner.kisters.de:1099/ClusterNodeControl20	Release_TSM_3-40-7 31.10.2013 14:47:54 CET
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
5001	kilosys	KIOSystem	7446	on	on	0 (0)	0		true		
4001	tsmsrver	tsmsrver	7445	on	on	0 (0)	0		clientGroup == "kilosys"	rmi://kn-jheimbuchner.kisters.de:1099/Tsm2	
3001	102001	kn-jheimbuchner.kisters.de	30	tsm-1	on	on	13.11.2013 13:02:20 CET	2	0.00	rmi://kn-jheimbuchner.kisters.de:1099/ClusterNodeControl30	Release_TSM_3-40-7 31.10.2013 14:47:54 CET
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
2001	tsmsrver	tsmsrver	7443	on	off	9 (9)	0		true	rmi://kn-jheimbuchner.kisters.de:1099/Tsm1	
4001	103001	kn-jheimbuchner.kisters.de	40	interactive	on	on	13.11.2013 13:02:26 CET	2	0.00	rmi://kn-jheimbuchner.kisters.de:1099/ClusterNodeControl40	Release_TSM_3-40-7 31.10.2013 14:47:54 CET
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
6001	operationserver	OperationServer	7447	on	on	0 (0)	0		true		
3001	tsmsrver	tsmsrver	7444	on	on	0 (0)	0		clientGroup != "kilosys"	rmi://kn-jheimbuchner.kisters.de:1099/Tsm3	
7001	wiskiserver	WISKIServer	7448	on	on	0 (0)	0		true		
5001	106001	kn-jheimbuchner.kisters.de	50	scriptserver	on	on	13.11.2013 13:01:38 CET	2	0.00	rmi://kn-jheimbuchner.kisters.de:1099/ClusterNodeControl50	Release_TSM_3-40-7 31.10.2013 14:47:54 CET
Applications in Cluster Node											
Key	Type	Name	IP Port	State	Interactive Usage	Sessions	Tasks	Assigned Tenants	Select Expression	RMI URL	
8001	scriptserver	scriptserver	7449	on	on	0 (0)	0		true		

KISTERS

install user Cluster Admin

AdminDomain

13.11.2013 13:02:36 CET

KISTERS

install user Cluster Admin kiAdminDomain 13.11.2013 13:02:38 CET

In the steps before the initialisation routine created the tenant database schema. This database schema contains the KiTSM and Dispatcher tables as well as the WISKI7 database tables. The WISKI7 tables are generated through **DbMaintain.exe**. This tenant schema must be **ACTIVE** for the Dispatcher. If the desired tenant is not **ACTIVE**, WISKI7 returns a **NO OPEN TENANT DATABASE** error while logging in to KiTSM.

TENANTS > TENANTS

Home

TSM Cluster

Log Viewer

Tenants

Authorisation

Preferences

Application Db Plugins

Groups

Tenants

1 - 1 / 1

Id	Tenant	State	Session Duration	Groups	Template	Auth. Type Key
	Id	Database storage	State	Extra db	JDBC URL	Db user
	Id	Application-db	State	Exists	Version up to date	Version
1001	MaDomain	online	900			
1001	tsmDb	online	no	jdbc:sqlserver://kn-jheimbuchner\SQLEXPRESS:1433;DatabaseName=wiski		wiskiadm
1001	tsmsrver	available	yes	yes	10_11	yes
2001	kilosys	available	yes	yes	2_09	yes
2001	tcaExDb	online	yes	jdbc:sqlserver://kn-jheimbuchner\SQLEXPRESS:1433;DatabaseName=wiski		wiskiadm

Congratulations! Your KISTERS WISKI server has been successfully installed.

10.2 WISKI7 Client Installation Tasks

WISKI7 as the client application of KiTSM can be installed on Terminal servers or on workstations. You can install WISKI7 manually, using an Install Shield GUI, in silent mode or using console commands. There is no difference in the installation procedure, thus the following guide can be applied for both installation types. The actual time series data will be accessed directly through the KiTSM API. Getting KiTSM data does not require an RDBMS client installed locally. However, WISKI7 still requires a RDBMS client for the classical Meta Data Management. Consider the installation requirements for the WISKI7 client application.

The client installation tasks comprise the following steps:

- [Initialising the WISKI7 Database](#) ^[104]

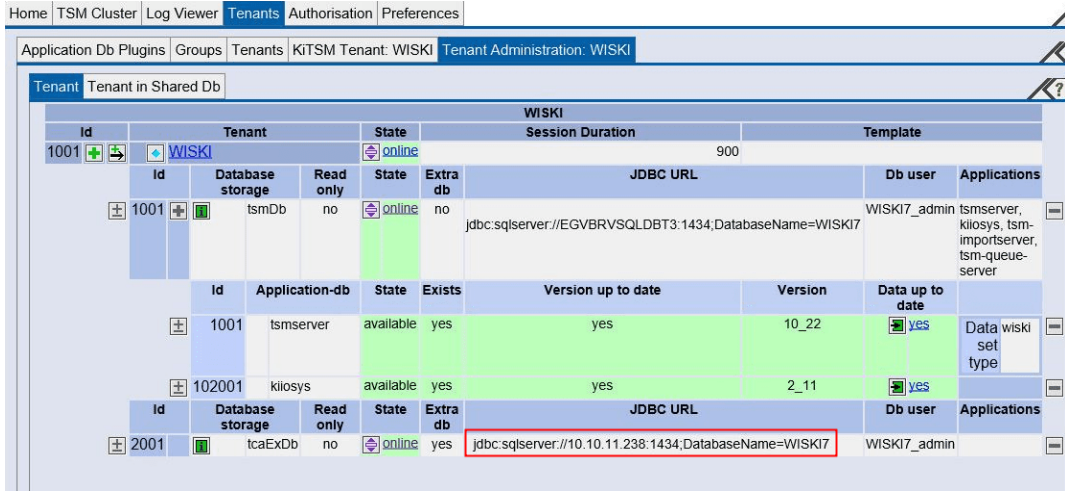
10.2.1 Initialising the WISKI7 Database

Navigate with the WINDOWS explorer to <root>\wiski\bin.

Execute `wiski_client_update.bat`, fill out the connection data depending on your system installation.

The **DATABASE MAINTENANCE** wizard (also known as `DbMaintain.exe`) opens (see screenshots below).

Element	Description
User	Enter the schema name of the database user here.
Password	Enter the password here.
Database	Enter the schema name of the database user here.
Server	Enter the server here.

Element	Description
	<p>Note: You can also use the port configuration (besides <code><server name>\<instance name></code>) when connecting to an MSSQL database. Separate the port number by comma from the URL or IP address.</p> <p>Example: <code>10.10.11.238, 1434</code>. This entry matches the following example KiTSM setup:</p> 
Connection type	Select the database type and version here.

Database maintenance



Database connection

User:
Database:
Server:
Connection type:

Configuration of application module(s) to be installed / maintained

☒ WISKI / KiTSM
☒ BIBER / KWQM
☐ SGM (Metadata of hydrological stations)
☐ WSP-Fix

< Back

Next >

Cancel

Help

Element	Description
User	The database user is displayed here.
Database	The database is displayed here.

Element	Description
Server	The server is displayed here.
Connection type	The connection type is displayed here.
Configuration of application module(s) to be installed / maintained	
WISKI / KiTSM	Check this box to install / update the WISKI core module.
BIBER / KiWQM	Check this box to install / update the BIBER and KiWQM module.
SGM (Metadata of hydrological stations)	Check this box to install / update the SGM module.
WSP-Fix	Check this box to install / update the WSP-Fix module.

Database maintenance

Database connection

User: w7411docuen

Database: w7411docuen

Server: orcln

Connection type: OracleOCI

Configuration

Database language: English

☐ Create database

☒ Update database

Tablespace for new objects

For tables: W7411DOCUEN

For indices: W7411DOCUEN

< Back Next > Cancel Help

Element	Description
User	The database user is displayed here.
Database	The database is displayed here.
Server	The server is displayed here.
Connection type	The connection type is displayed here.
Configuration	

Element	Description
Database language	Specify the language of the database here. Note: This is only possible when you initialise the database. When updating the database, the language cannot be changed.
Create database	Check this box to create the database. Note: This is only possible when you initialise the database.
Update database	Check this box to update the database. Note: This is only possible when you update the database.
Tablespace for new objects	
For tables	Specify optionally the tables for which you want to create the tablespace for new objects here.
For indices	Specify optionally the indices for which you want to create the tablespace for new objects here.

Database maintenance

Database connection

User: w7411docuen

Database: w7411docuen

Server: orcln

Connection type: OracleOCI

Additional (optional) data maintenance configuration

Optional Mandatory

- ☐ Check all role permissions on TCA with default values
- ☐ Reset all assignments between tables and functional groups to default
- ☒ Check all table structures
- ☒ Update statistics in database
- ☒ Check trigger validity
- ☒ Create resp. update metrics metadata and time series
- ☒ Synchronise unit groups between TCA and KiTSM
- ☒ Synchronise parameter types between TCA and KiTSM
- ☒ Synchronise metadata between TCA and KiTSM
- ☒ Update TCA time zone historical infos from KiTSM
- ☒ Update of TCA roles atomic permissions on KiTSM
- ☒ Recreate system standard remarks

< Back Next > Cancel Help

Element	Description
User	The database user is displayed here.
Database	The database is displayed here.
Server	The server is displayed here.
Connection type	The connection type is displayed here.

Database maintenance

Database connection

User: w7411docuen

Database: w7411docuen

Server: orcln

Connection type: OracleOCI

Additional (optional) data maintenance configuration

Optional Mandatory

- ☐ Check all role permissions on TCA with default values
- ☐ Reset all assignments between tables and functional groups to default
- ☒ Check all table structures
- ☒ Update statistics in database
- ☒ Check trigger validity
- ☒ Create resp. update metrics metadata and time series
- ☒ Synchronise unit groups between TCA and KiTSM
- ☒ Synchronise parameter types between TCA and KiTSM
- ☒ Synchronise metadata between TCA and KiTSM
- ☒ Update TCA time zone historical infos from KiTSM
- ☒ Update of TCA roles atomic permissions on KiTSM
- ☒ Recreate system standard remarks

< Back Next > Cancel Help

Additional (optional) data maintenance configuration

Optional

Check all role permissions on TCA with default values	
Reset all assignments between tables and functional groups to default	Check this box to reset all assignments between tables and functional groups to default.
Check all table structures	Check this box to check all table structures.
Update statistics in database	Check this box to update statistics in the database.
Check trigger validity	Check this box to evaluate the user-specific triggers for ORACLE databases for validity. Note: Besides, an additional task to check if all triggers are valid was added to the TCA database maintenance.
Create resp. update metrics metadata and time series	
Synchronise unit groups between TCA and KiTSM	Check this box to synchronise the unit groups between the WISKI client and KiTSM.
Synchronise parameter types between TCA and KiTSM	Check this box to synchronise the parameter types between the WISKI client and KiTSM.
Synchronise metadata between TCA and KiTSM	Check this box to synchronise the metadata between the WISKI client and KiTSM.

Additional (optional) data maintenance configuration

Update TCA time zone historical infos from KiTSM	Check this box to update the WISKI client's historical information of the time zone from KiTSM.
Update of TCA roles atomic permissions on KiTSM	Check this box to update the WISKI client's roles atomic permissions on KiTSM.
Recreate system standard remarks	Check this box to recreate system standard remarks.

Database maintenance

Database connection

User: w7411docuen

Database: w7411docuen

Server: orcln

Connection type: OracleOCI

Additional (optional) data maintenance configuration

Optional Mandatory

- ☒ Fill/check assignment between functional groups and atomic permissions
- ☒ Portal admin user creation
- ☒ Check all primary keys and indices
- ☒ Check all table to functional group assignments
- ☒ Fill/check system parameter types
- ☒ Maintain IO Metadata
- ☒ Fill/check system time series types
- ☒ Fill/check mapping between time series types and agent types.
- ☒ Fill missing report instance type XML configuration
- ☒ Fill missing default report instance types
- ☒ Fill/check system time series parameters on KiTSM
- ☒ Creating system time series

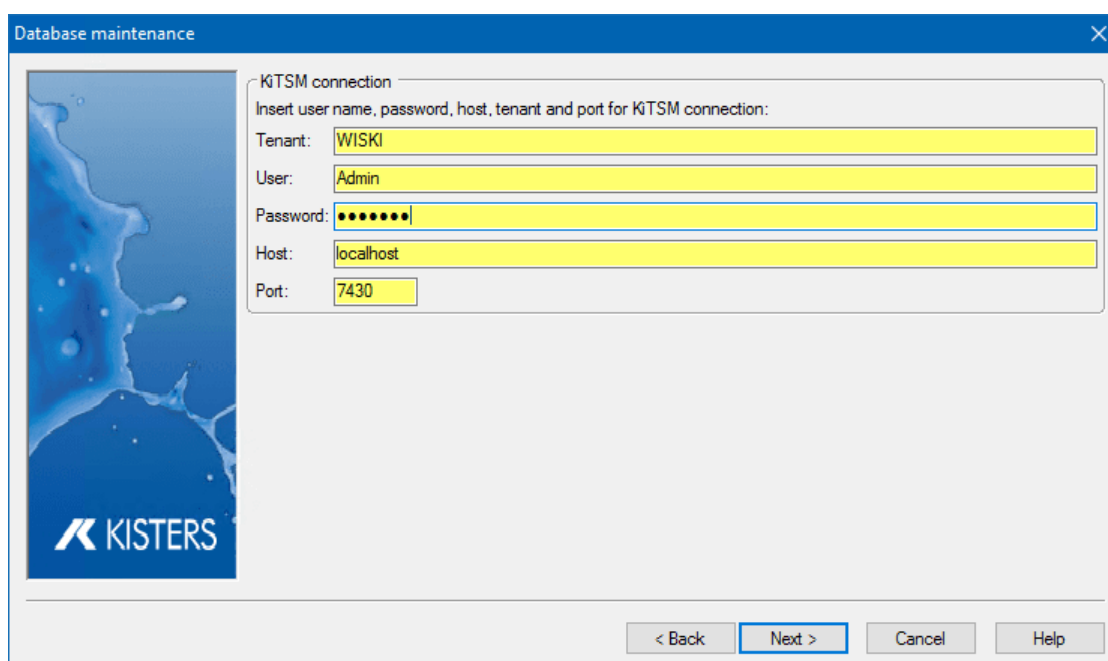
< Back Next > Cancel Help

Additional (optional) data maintenance configuration**Mandatory**

Fill/check assignment between functional groups and atomic permissions	
Portal admin user creation	
Check all primary keys and indices	
Check all table to functional group assignments	
Fill/check system parameter types	
Maintain IO Metadata	
Fill/check system time series types	

Additional (optional) data maintenance configuration

Fill/check mapping between time series types and agent types	
Fill missing report instance type XML configuration	
Fill missing default report instance types	
Fill/check system time series parameters on KiTSM	
Creating system time series	



Database maintenance

KiTSM connection

Insert user name, password, host, tenant and port for KiTSM connection:

Tenant:

User:

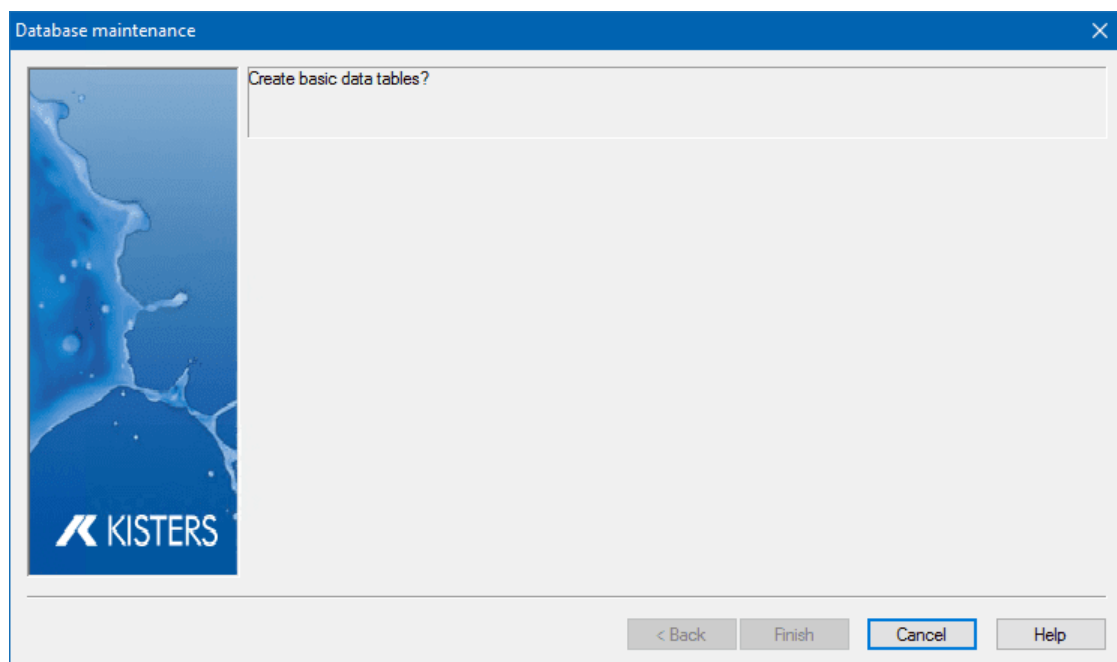
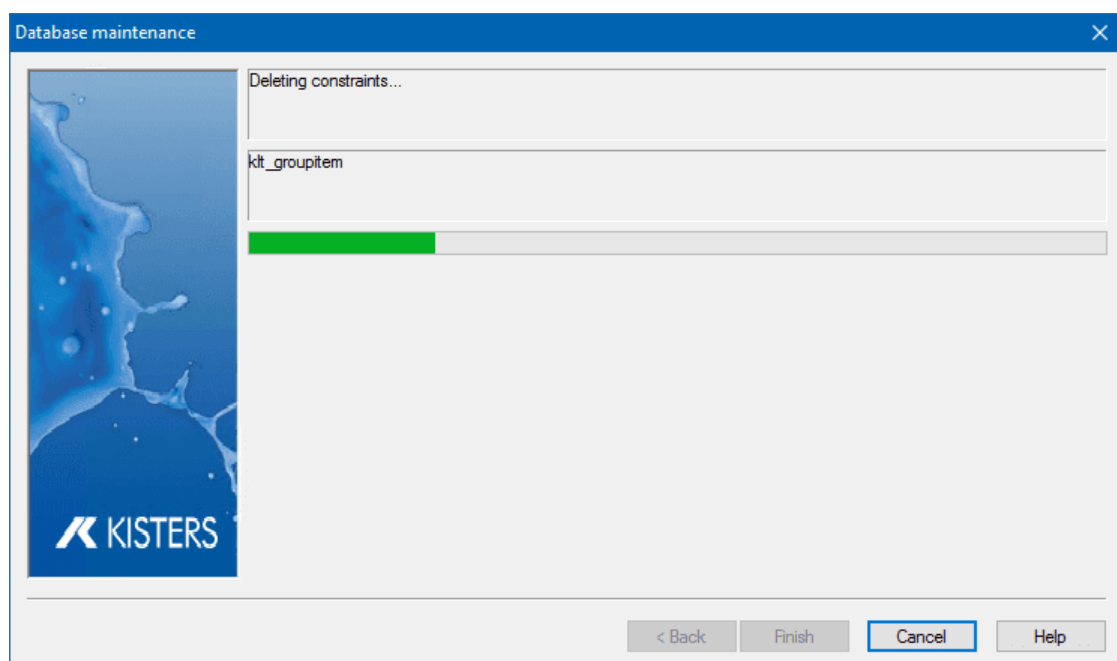
Password:

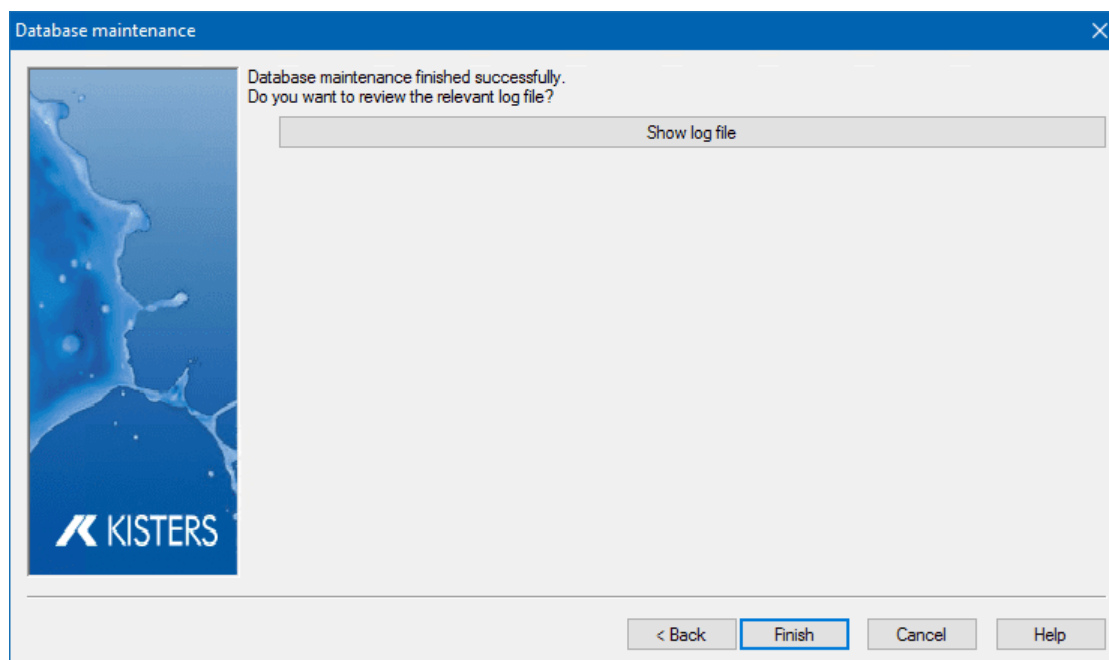
Host:

Port:

< Back Next > Cancel Help

Element	Description
Tenant	Enter the tenant for the KiTSM connection here.
User	Enter the user name for the KiTSM connection here.
Password	Enter the password for the KiTSM connection here.
Host	Enter the host for the KiTSM connection here.
Port	Enter the port for the KiTSM connection here.





Element	Description
Show log file	Click this button to open the log file.

Note: While the database maintenance is running, the following tasks are performed in the background:

- Constrains are deleted and new created
- Views are deleted and new created
- Missing tables are created
- Database tables are updated
- Procedures, functions and packages are deleted and new created
- PK's und Indexes are corrected
- Validity of ORACLE triggers are checked
- Updating database statistics

10.3 Post-installation Steps

Changing Passwords of Default Users

Each new database is equipped with default users. A standard WISKI7 installation always includes the following default users:

- Admin
- Power_User (locked by default)
- User (locked by default)
- Guest (locked by default)

We recommend to change the passwords for these users at first use.

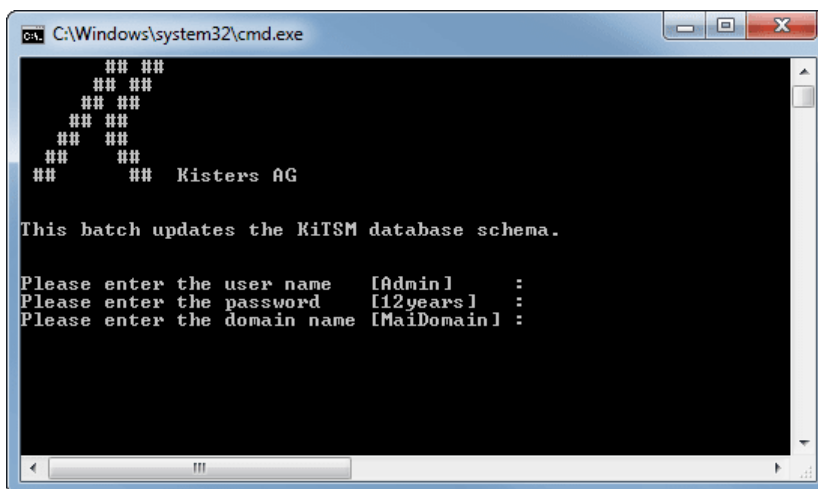
11 WISKI Server Update (Manual Update)

Put together a [package configuration](#). If, for example, you would like to install a 64-bit WISKI server installation, you need the following packages:

- WISKI-<vn>-<bn>-java-64bit.zip
- WISKI-<vn>-<bn>-kidsm.zip
- WISKI-<vn>-<bn>-kidsm-client-64.zip
- WISKI-<vn>-<bn>-kilog-client-64bit.zip
- WISKI-<vn>-<bn>-services.zip
- WISKI-<vn>-<bn>-wiski-client.zip
- WISKI-<vn>-<bn>-wiski-client-help-all.zip

Updating a previous installation with new directory structure

- Shut down and uninstall the services.
Create a backup of the existing configuration:
- Navigate to <root>\wiski\bin and execute the batch file **configuration_backup.bat**.
- This batch file will automatically backup your configuration and store all relevant data in <root>\backup.
Backup and install the previous installed software:
- Archive the wiski directory to <root>\previous_versions\WISKI<version number>_<build number>.zip.
- Delete the WISKI directory.
- Copy the package configuration into the root directory and unzip the files with the **EXTRACT HERE** context menu item.
Restore the configuration backup:
- Navigate to <root>\wiski\bin and execute the batch file **configuration_restore.bat**. This batch file will automatically restore your configuration from <root>\backup.
Execute the services update procedure:
- Navigate to <root>\wiski\bin and execute the batch file **wiski-server_update_db.bat**.



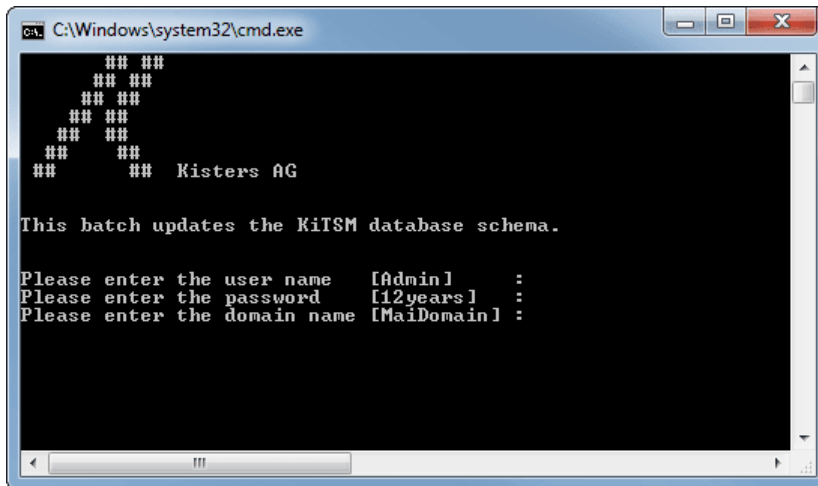
- The batch file will ask for the WISKI login data. The default values are displayed in brackets. If the default values are to be used, click **[RETURN]**.
Execute the client update procedure:
- Navigate to <root>\wiski\bin and execute the batch file **wiski_client_update.bat**.

11.1 Updating a previous installation with new directory structure

- Shut down and uninstall the services.
Create a backup of the existing configuration:
- Navigate to <root>\wiski\bin and execute the batch file **configuration_backup.bat**.
- This batch file will automatically backup your configuration and store all relevant data in <root>\backup.
Backup and install the previous installed software:
- Archive the wiski directory to <root>\previous_versions\WISKI<version number>_<build number>.zip.
- Delete the WISKI directory.
- Copy the package configuration into the root directory and unzip the files with the **EXTRACT HERE** context menu item.
Restore the configuration backup:
- Navigate to <root>\wiski\bin and execute the batch file **configuration_restore.bat**. This batch file will automatically restore your configuration from <root>\backup.

Execute the services update procedure:

- Navigate to `<root>\wiski\bin` and execute the batch file `wiski-server_update_db.bat`.



- The batch file will ask for the WISKI login data. The default values are displayed in brackets. If the default values are to be used, click **RETURN**.
- Execute the client update procedure:
- Navigate to `<root>\wiski\bin` and execute the batch file `wiski_client_update.bat`.

12 Setting Up a Satellite KiTSM Server

Setting up a satellite KiTSM server mainly comprises these steps:

- [Setting Up a Calculation Node for Satellites](#)^[115]
- [Setting Up an Interactive Node for Satellites](#)^[117]

12.1 Setting Up a Calculation Node for Satellites

Setting up a calculation node for the satellite KiTSM server comprises these steps:

- [Required packages for Satellites](#)^[115]
- [Since WISKI 7.4.3 SR12](#)^[115]
- [Before WISKI 7.4.3 SR12](#)^[116]

12.1.1 Required packages for Satellites

- WISKI-<vn>-<bn>-java-64bit.zip OR WISKI-<vn>-<bn>-java-32bit.zip
- WISKI-<vn>-<bn>-services.zip

Create a root directory, e.g. `c:\kisters`. Place both packages in the folder and extract the files there.

12.1.2 Since WISKI 7.4.3 SR12

In the 7.4.3 SR13 version a major change in the cluster node setup has been made. Configuration is now XML-based and much simpler.

The procedure now consists of these steps:

- [Modifying the XML File](#)^[115]
- [Creating the Cluster Nodes](#)^[116]
- [Redirecting the Cluster Nodes to the Dispatcher](#)^[116]
- [Starting the Satellite Server](#)^[116]
- [Configuring the KiTSM Cluster](#)^[116]

12.1.2.1 Modifying the XML File

The cluster node profile file can be found in the `<root>\wiski\services\WISKIServer_ClusterProfile.xml` file.

- Open the file and scroll to the bottom.
- Remove the **dcsInstance** with the names: `tsmdispatcher`, `queueserver`, `automation`, `interactive` and `scriptserver`.
- Change the name from **tsm1** to the next free name, e.g. **tsm2**.
Note: You check the next free name on the cluster node page of the KiTSM web interface.
- Change the ID from 30 to the next free ID, e.g. 60.
Note: You check the next free name on the cluster node page of the KiTSM web interface.
- Set the **maxMemory** field to the amount of memory you would like to set up for the Java virtual machine.
- Remove the **jprocService** node with the name **dispatcherdb**.

Example:

```
...
<!-- INSTALLATION CLUSTER STRUCTURE -->
<!-- DCServer Instances -->
<dcsInstances>
  <dcsInstance name="tsm2" type="tsm" default="true" id="60">
    <maxMemory>4096</maxMemory>
    <jprocSequenceNumber>80</jprocSequenceNumber>
  </dcsInstance>
</dcsInstances>
<!-- NON-OSGi Kisters modules, KiDSM, KiLogServer, Dcs Instances
will get corresponding JProc service configurations automatically.
-->
```

```
<jprocServices>
  <jprocService name="kilogserver" sequenceNumber="06"
moduleDir="LogServer" />
</jprocServices>
</KiCluster>
```

12.1.2.2 Creating the Cluster Nodes

- Navigate to the directory `<root>\wiski\services\tsm\bin` and execute the Perl script `createOrUpdateDcsServer.pl`.

```
> perl createOrUpdateDcsServer.pl
```

12.1.2.3 Redirecting the Cluster Nodes to the Dispatcher

By default the cluster nodes expect the dispatcher to be on the local host. You must change this to the correct computer where the dispatcher is running.

- Navigate to `<root>\wiski\services\dcserver\server_tsm1\configuration`.
- Create a new file with the name `dcsmgr-cluster-nodecontroller-custom.properties` with the following content:

```
tsms.dispatcher.rmiurl=rmi://localhost:1099/TsmDispatcher
```

- Replace `localhost` with the computer name or the IP address of the dispatcher server.

12.1.2.4 Starting the Satellite Server

- Navigate to `<root>\wiski\services\jprocontroller\bin` and execute the `jprocontroller.pl` script

```
> perl jprocontroller.pl -daemon
```

12.1.2.5 Configuring the KiTSM Cluster

- Set **INTERACTIVE USAGE** of the new server to **OFF**.
- Set **ADM STATE** of the new server to **ON**.
- Configure background processes for node 60.
- Set up the server as a service.

12.1.3 Before WISKI 7.4.3 SR12

Setting up a calculation node for the satellite KiTSM server for versions prior to WISKI 7.4.3 SR12 involves the following steps:

- [Setting up the JProcController](#)^[116]
- [Setting up the DCServer](#)^[117]
- [Registering the calculation modules](#)^[117]
- [Starting the satellite server](#)^[117]
- [Configuring the KiTSM cluster](#)^[117]

12.1.3.1 Setting up the JProcController

- Navigate to `<root>\wiski\services\jprocontroller\configuration`.
- Rename the file `80-jproc-tsm1-default.properties` to `90-jproc-tsm2-default.properties`.
- Edit `90-jproc-tsm2-default.properties`, search and replace all `tsm-1` to `tsm-2` and `tsm1` to `tsm2`.
- Delete all other files in `<root>\wiski\services\jprocontroller\configuration` that start with a number.

12.1.3.2 Setting up the DCServer

- Navigate to `<root>\wiski\services\dcserver`.
- Enter `create-dcserverinstance.pl tsm2 1024` in the command prompt and execute.
The second parameter value is the amount of memory which should be assigned.
- Copy all files from `<root>\wiski\services\dcserver\server_tsm1\configuration` to `<root>\wiski\services\dcserver\server_tsm2\configuration`.
- Edit all copied files, search and replace all `tsm-1` by `tsm-2` and all `tsm1` by `tsm2`.
- Edit the file `<root>\wiski\services\dcserver\server_tsm2\configuration\dcsmgr-cluster-nodecontroller-default.properties`.
- Change `cluster.node.instance=30` to `cluster.node.instance=60`
Note: This depends on the last instance, which can be looked up in the KiTSM web interface.
- Change `tsms.dispatcher.rmiurl=rmi://localhost:1099/TsmDispatcher` to `tsms.dispatcher.rmiurl=rmi://<Server where the dispatcher is running>:1099/TsmDispatcher`.
- Rename `cluster-node#30-log4j.properties` to `cluster-node#60-log4j.properties`.

12.1.3.3 Registering the calculation modules

When you are working with a WISKI version 7.3:

- Navigate to `<root>\wiski\services\wiski-server` and execute `perl install.pl -c wiski-agents tsm2`.

When you are working with a WISKI version 7.4 or higher:

- Navigate to `<root>\wiski\services\tools` and execute `perl install.pl wiski-server tsm2 -c wiski-agents`.

12.1.3.4 Starting the satellite server

Navigate to `<root>\wiski\services\jprocontroller\bin` and execute `jprocontroller.pl -daemon`.

12.1.3.5 Configuring the KiTSM cluster

- Set **ADM STATE** of the new server to **ON**.
- Set **INTERACTIVE USAGE** of the new server to **OFF**.
- Configure background processes for node 60.
- Set up the server as a service.

12.2 Setting Up an Interactive Node for Satellites

Setting up an interactive node for the satellite KiTSM server comprises these steps:

- [Required packages for an Interactive Node](#)^[117]
- [Since WISKI 7.4.3 SR12](#)^[117]
- [Before WISKI 7.4.3 SR12](#)^[119]

12.2.1 Required packages for an Interactive Node

- WISKI-`<vn>-<bn>-java-64bit.zip` OR WISKI-`<vn>-<bn>-java-32bit.zip`
- WISKI-`<vn>-<bn>-services.zip`

Create a root directory, e.g. `c:\kisters`. Place both packages in the folder and extract the files there.

12.2.2 Since WISKI 7.4.3 SR12

In the 7.4.3 SR13 release, a major change has been made to the cluster node setup. The configuration is now XML-based and much simpler.

The procedure now consists of the following steps:

- [Modifying the XML File](#)^[118]

- [Creating the Cluster Nodes](#)^[118]
- [Redirecting the Cluster Nodes to the Dispatcher](#)^[118]
- [Starting the Satellite Server](#)^[119]
- [Configuring the KiTSM Cluster](#)^[119]

12.2.2.1 Modifying the XML File

The cluster node profile file can be found in the `<root>\wiski\services\WISKIServer_ClusterProfile.xml` file.

- Open the file and scroll to the bottom.
- Remove the `dcsInstance` with the names: `tsmdispatcher`, `queueserver`, `automation`, `tsm1` and `scriptserver`.
- Change the name from `interactive` to the next free name, e.g. `interactive1`.
Note: You check the next free name on the cluster node page of the KiTSM web interface.
- Change the ID from `40` to the next free ID, e.g. `70`.
Note: You check the next free name on the cluster node page of the KiTSM web interface.
- Set the `maxMemory` field to the amount of memory you would like to set up for the Java virtual machine.
- Remove the `jprocService` node with the name `dispatcherdb`.

Example:

```
...

<!-- INSTALLATION CLUSTER STRUCTURE -->
<!-- DCServer Instances -->
<dcsInstances>
  <dcsInstance name="interactive2" type="interactive"
default="true" id="70">
    <maxMemory>4096</maxMemory>
    <jprocSequenceNumber>60</jprocSequenceNumber>
  </dcsInstance>
</dcsInstances>
<!-- NON-OSGi Kisters modules, KiDSM, KiLogServer, Dcs Instances
will get corresponding JProc service configurations automatically.
-->
<jprocServices>
  <jprocService name="kilogserver" sequenceNumber="06"
moduleDir="LogServer" />
</jprocServices>
</KiCluster>
```

12.2.2.2 Creating the Cluster Nodes

- Navigate to the directory `<root>\wiski\services\tsm\bin` and execute the Perl script `createOrUpdateDcsServer.pl`.

```
> perl createOrUpdateDcsServer.pl
```

12.2.2.3 Redirecting the Cluster Nodes to the Dispatcher

By default the cluster nodes expect the dispatcher to be on the local host. You must change this to the correct computer where the dispatcher is running.

- Navigate to `<root>\wiski\services\dcserver\server_interactive2\configuration`.
- Create a new file with the name `dcsmgr-cluster-nodecontroller-custom.properties` with the following content:

```
tsms.dispatcher.rmiurl=rmi://localhost:1099/TsmDispatcher
```

- Replace `localhost` with the computer name or the IP address of the dispatcher server.

12.2.2.4 Starting the Satellite Server

- Navigate to `<root>\wiski\services\jprocontroller\bin` and execute the `jprocontroller.pl` script

```
> perl jprocontroller.pl -daemon
```

12.2.2.5 Configuring the KiTSM Cluster

- Set **INTERACTIVE USAGE** of the new server to **OFF**.
- Set **ADM STATE** of the new server to **ON**.
- Configure background processes for node 60.
- Set up the server as a service.

12.2.3 Before WISKI 7.4.3 SR12

Setting up an interactive node for the satellite KiTSM server for versions prior to WISKI 7.4.3 SR12 involves the following steps:

- [Setting up the JProcController](#)^[119]
- [Setting up the DCServer](#)^[119]
- [Register the application server modules](#)^[119]
- [Start the satellite server](#)^[120]
- [KiTSM cluster configuration](#)^[120]

12.2.3.1 Setting up the JProcController

Navigate to `<root>\wiski\services\jprocontroller\configuration`.

- Rename the file `60-jproc-interactive-default.properties` to `90-jproc-interactive-default.properties`.
- Edit `90-jproc-interactive-default.properties`, search and replace all `interactive-2` to `interactive-2`.
- Delete all other files which start with a number in `<root>\wiski\services\jprocontroller\configuration`.

12.2.3.2 Setting up the DCServer

Navigate to `<root>\wiski\services\dcserver` and execute in the command line `create-dcserverinstance.pl interactive-2 1024`. The second parameter is the amount of memory which should be assigned.

- Copy all files from `<root>\wiski\services\dcserver\server_interactive\configuration` to `<root>\wiski\services\dcserver\server_interactive-2\configuration`. Edit all copied files, search and replace all `interactive` to `interactive-2`.
- Edit the file `<root>\wiski\services\dcserver\server_interactive-2\configuration\dcsmgr-cluster-nodecontroller-default.properties` and change two entries:
 - `cluster.node.instance=30` to `cluster.node.instance=60` [

Note: This depends on the last instance, which can be looked up in the KiTSM webinterface.
 - `tsms.dispatcher.rmiurl=rmi\://localhost\:1099/TsmDispatcher` to `tsms.dispatcher.rmiurl=rmi\://<Server>\:1099/TsmDispatcher` where the dispatcher runs.
- Rename `cluster-node#40-log4j.properties` to `cluster-node#60-log4j.properties`.

12.2.3.3 Register the application server modules

- Navigate to `<root>\wiski\services\wiski-server` and execute the following commands:
 - `install.pl -c wiski-server interactive-2`
 - `install.pl -c wiski-operations interactive-2`
- Navigate to `<root>\wiski\services\operationframework-server` and execute the following command:
 - `install.pl -c operationframework-server interactive-2`

12.2.3.4 Start the satellite server

Navigate to `<root>\wiski\services\jprocontroller\bin` and execute `jprocontroller.pl -daemon`.

12.2.3.5 KiTSM cluster configuration

- Set the **ADM** state of the new server to **ON**.
- Set the **SELECT EXPRESSION** of the KiTSM of the new node to `clientGroup != "kiosys"` like the origin interactive node.
- Set up the server as a service.

13 Linux Installation

When installing WISKI under Linux, read the following sub chapters:

- [Initial Installation](#) 
- [Updating an Existing Installation](#) 

13.1 Initial Installation

Copy the following packages into a folder to be used for installation, e.g. /kisters/:

- WISKI-<vn>-<bn>-java-64bit-linux.tar.gz
- WISKI-<vn>-<bn>-jprocontroller-client-64bit-linux.tar.gz
- WISKI-<vn>-<bn>-jprocontroller-client-default-config.zip
- WISKI-<vn>-<bn>-services.zip

Step	Description	Command
1	Extract all tar.gz files.	<pre>for i in *.tar.gz; do tar -xvzf \$i; done</pre>
2	Extract all ZIP archives.	<pre>unzip '*.zip'</pre>
4	Navigate to the management directory.	<pre>cd wiski/bin/</pre>
5	Execute the configuration script.	<pre>./configureLinux.sh</pre>
6	Execute the init script, fill in all information and run the initialisation process.	<pre>./wiski-server_initialize_db.sh</pre>
7	Start the JProcController console to check all processes.	<pre>./jprocontroller_console.sh</pre>
8	Extract the WISKI client packages on a WINDOWS machine and execute DbMaintain.exe.	<pre>DbMaintain.exe</pre>

13.2 Updating an Existing Installation

Step	Description	Command
1	Shut down the old installation (KiDSM and KiTSM).	
2	Extract all packages. Note: The packages contain a wiski subdirectory. If the directory where you extract the files also contains a directory with the same name, rename it first.	

Step	Description	Command
3	Extract all tar.gz files.	<pre>for i in *.tar.gz; do tar -xvzf \$i; done</pre>
4	Extract all ZIP archives.	<pre>unzip *.zip</pre>
5	Copy the dispatcher.properties to <root>/wiski/services/dcserver/server_tsmdispatcher/configuration and name it: dcsmgr-dispatcher-custom.properties .	
6	Copy the kidsm-custom.properties to new installation: <root>/wiski/services/kidsm/kidsm-custom.properties (also migrate all custom specific converters).	
7	Navigate to the management directory <root>/wiski/bin .	<pre>cd wiski/bin/</pre>
8	Execute the configuration script.	<pre>./configureLinux.sh</pre>
9	Execute the services update script.	<pre>./wiski-server_update_db.sh</pre>
10	Execute the KiDSM update script.	<pre>./kidsm_update_db.sh</pre>
11	Start the JProcController console to check all processes.	<pre>./jprocontroller_console.sh</pre>
12	Extract the WISKI client packages on a WINDOWS machine and execute DbMaintain.exe .	<pre>DbMaintain.exe</pre>

14 Certificate and TLS Management in WISKI

Technical Background Information

HTTPS is an extension of the HTTP protocol, which allows to encrypt the communication channels using TLS and to authenticate via digital certificates. See [HTTPS](#) for more information.

Transport Layer Security (TLS) is a cryptographic protocol designed to provide communications security over a computer network. See [Transport Layer Security](#) for more information.

There are different protocol versions: TLS v1.0 and v1.1 are deprecated. TLS v1.2 is the current default in WISKI Server, while TLS v1.3 is supported but still not fully tested by QA and not the default right now.

There is also SSL in version v1, v2 and v3 - all 3 are deprecated and insecure. TLS is the successor specification, but many people still refer to SSL instead of TLS.

WISKI Server does not allow communication with any protocol version older than TLS v1.2.

Digital certificates are electronic documents used to prove the validity of a public key. See [Public key certificate](#) for more information.

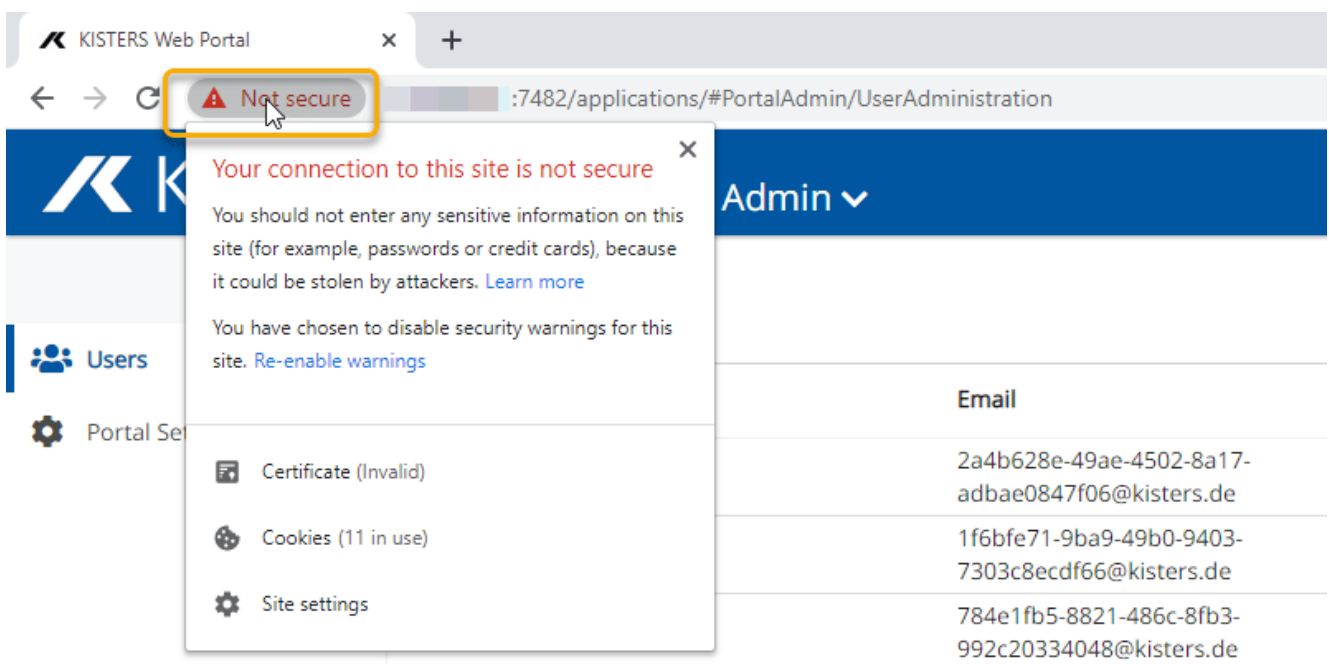
In general, the server encrypts communication using its secret private key, while the client uses the public key to decrypt and vice-versa. To be able to trust the server as being the one we expect it is, certificates are being used. The server provides a server certificate (its public key with additional metadata) to allow to check its identity.

In simple words: an administrator created a private key and Certificate Signing Request (CSR) for its server, asking a trusted root certification authority (CA, e.g. VeriSign, Microsoft, ...) to validate its identity based on provided CSR metadata. Once validated, the root CA will sign the server certificate with its secret private key and provide the signed server certificate back to the administrator. Once the administrator configured its server with the secret private key and the signed server certificate, each client will be able to check the server certificate for validity: e.g. checking the expiration time of the certificate, the hostname it belongs to and finally, if it's signed by a trusted root CA so that the client knows that the server can be trusted.

Certificates in WISKI

The default delivery of the WISKI7 Standard Portal works with a **KISTERS self-signed certificate**, which can be used in any target infrastructure. This provides secured access to the portal with any server name. Unfortunately, current browsers indicate an insecure warning in the title bar. This is because the free KISTERS certificate is not registered with an official certificate authority (CA).

- You can check this by clicking the insecure information in the title bar as follows:



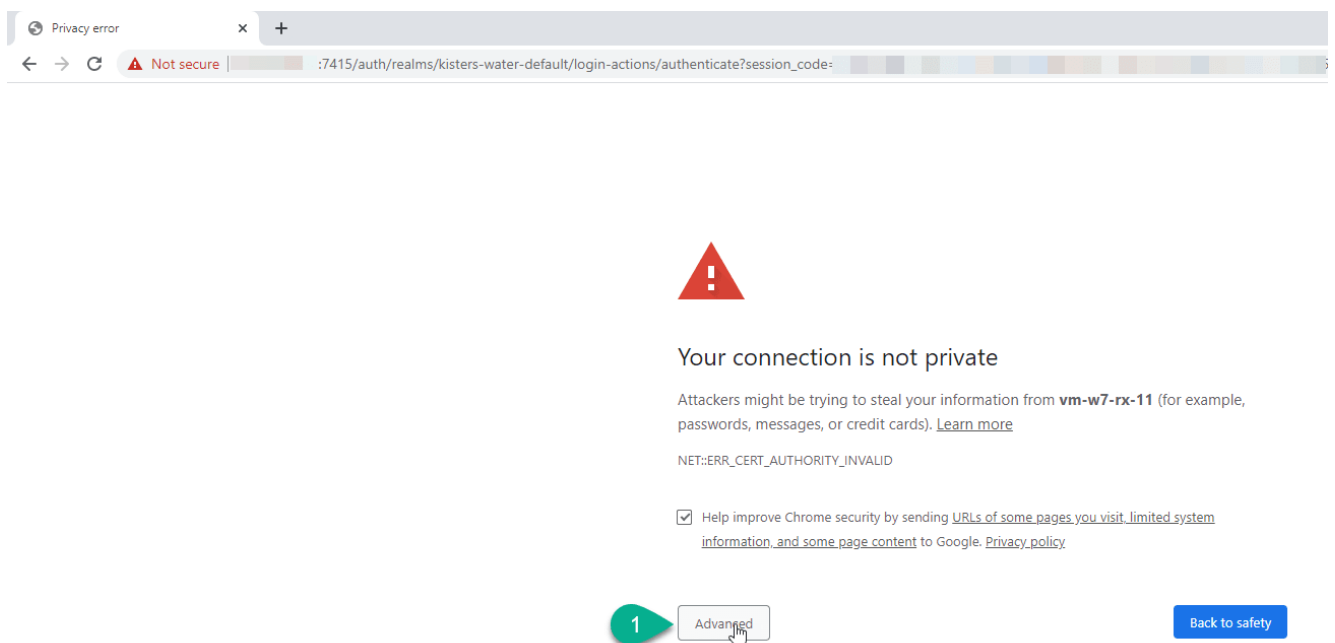
- To improve this behaviour, you can get in touch with your corporate IT department and look for a fully qualified certificate for your server.

This is typically valid for all domains of your organisation, for instance for *.[kisters.de](https://www.kisters.de). Alternatively, you can purchase a certificate for the desired domain name (you might look for GlobalSign, GoDaddy, Namecheap, [SSL.com](https://www.ssl.com), DigiCert, Thawte or others).

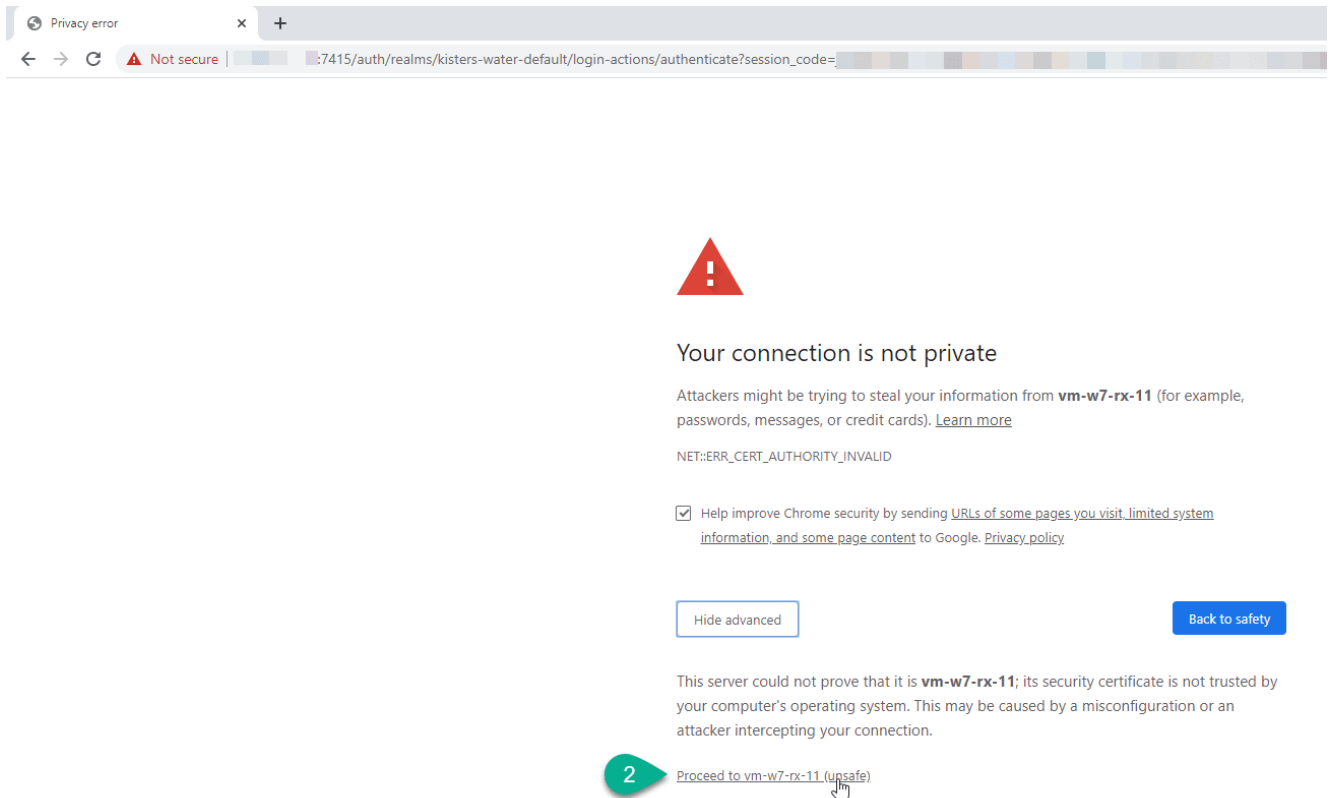
- You can substitute the default certificate using the installation/update wizard dialog of the WISKI Server Manager:

While the KISTERS certificate is active, users must accept the following dialog, which indicates that the connection COULD be insecure, which is not the case with this installation.

- Click the two steps to be able to proceed to the secure portal applications.
- Click **[ADVANCED]**.



- Click **[PROCEED TO...]**.



- To make your operating system trust the KISTERS certificate, it can be imported manually. The default certificates and keys can be found in **wiski/services/config/local/unmanaged/security**. Refer to your operating system manual on how to import certificates.

If you would like to use your own certificates, provide the following certificate data to KISTERS:

- X.509 Root Certification Authority Certificate
File format: PEM
- X.509 Server Certificate - including the whole certificate chain up to the root CA
File format: PEM
If and only if the server certificate has a X.509 Extended Key Usage extension: the purposes TLS Web Server Authentication and TLS Web Client Authentication (for RMI) must both be set!
- Server Certificate Private Key
File format: PEM
Private key format: PKCS #8 or PKCS #1
Private key algorithm: EC or RSA

Read the chapters hereafter for detailed information on certificate and TLS management in WISKI:

- [WISKI Server Configuration](#)^[125]
- [WISKI Client Configuration](#)^[127]
- [WISKI Portal Configuration](#)^[127]
- [Security Monitor](#)^[127]

Also note the [FAQ on TLS and Certificate Management](#)^[228].

14.1 WISKI Server Configuration

WISKI Server has one main configuration file for everything related to TLS and certificates: **services/config/local/security-default.properties**.

This file is available on each machine WISKI Server is installed on (no matter whether it is a master or a [satellite](#)^[115] node).

The configuration file allows to override the location of the WISKI Server KeyStore, its password and the KeyStore entry alias used for WISKI Server.

Also, it allows to specify the location of the WISKI Server TrustStore and its password.

All configuration items are documented in the property file itself. In case you will decide to override some settings, remind to create a custom configuration file to do so.

The KeyStore and TrustStore passwords are encrypted. You can generate new encrypted password by using the [WISKI Server Security Tool](#)^[154].

This configuration file is consumed by the WISKI Server Security Tool to set up all other required configurations in WISKI Server. Components like Keycloak, Tomcat, DCServer, Karaf PAX Web instances, etc., all require to be configured for TLS. To avoid having to configure the KeyStore/TrustStore locations again and again for all endpoints, the Security Tool just evaluates our centralised configuration and applies the required configuration to each thirdparty library or tool.

Beside the central configuration file, there are more files involved in the TLS or certificates support for WISKI Server. By default, they are all stored in **services/config/local/unmanaged/security**:

File	Usage
rootCA.crt	<p>The self-signed root Certification Authority (CA) certificate.</p> <p>Should only be available on systems using self-signed certificates. Systems using official certificates don't need an own root CA. It's typically also stored in the WISKI Server TrustStore.</p>
rootCA.key	<p>The root CA private key.</p> <p>It should only be available on systems using self-signed certificates. Systems using official certificates don't need their own root CA.</p> <p>It's confidential and should never be shared, put in unprotected folders etc. - this applies especially to productive systems. If it leaks only 1 time, the certificate as well as the whole certificate chain (=> all certificates signed by this CA) must be considered insecure, revoked and be re-generated again.</p>
server.crt	<p>The self-signed server certificate, signed by the self-signed or official CA.</p> <p>Typically not available on filesystem as it has no real usage - use the server certificate chain instead of. WISKI Server will only write it when DEBUG is set.</p>
chain.crt	<p>The server certificate chain consumed by the server endpoints. It's always available on filesystem to simplify issue analysis but is consumed together with the server private key from within the WISKI Server KeyStore.</p>
server.key	<p>The server private key.</p> <p>Typically not available in filesystem as it's security-sensitive and anyways stored in the secured WISKI Server KeyStore. It will only be written on filesystem when DEBUG is set.</p> <p>It's confidential and should never be shared, put in unprotected folders etc. - this applies especially to productive systems using official instead of self-signed certificates. If it leaks only 1 time, the certificate must be considered insecure, revoked and be re-generated (self-signed) or bought (official) again.</p>
keystore.p12	<p>The KeyStore is password protected and contains the server certificate chain and its private key.</p> <p>It's typically the preferred way to access the server certificate and private key by our services.</p> <p>It's confidential and should never be shared, put in unprotected folders etc.</p>
truststore.p12	<p>The TrustStore (technically also a KeyStore) is also password protected and contains CA certificates (e.g. the KISTERS self-signed CA certificate or a customer CA certificate).</p> <p>It's typically the preferred way to access the trusted CA certificates by our services. It will be merged with the Java default TrustStore at runtime.</p> <p>It's not confidential since it contains public certificates only</p>

Note that since WISKI 7.4.13 SR11, WISKI Server can also consume the Windows TrustStore - therefore it's not always required to configure the root CA certificate in our own TrustStore. It's important that:

- it's official and provided by the JDK TrustStore,
- it's not official and provided by the WISKI Server TrustStore,
- or that it's not official but provided by the Windows TrustStore.

Nevertheless, keep in mind that if some certificates are added by some "evil users" to this store, WISKI Server could start to trust remote endpoints it shouldn't trust.

14.2 WISKI Client Configuration

The WISKI Desktop does not require any additional configuration as client application, but there are some things to know:

14.3 WISKI Portal Configuration

WISKI Portal does not require any additional configuration, as everything is managed in the container - Tomcat in this case. The configuration of Tomcat is managed within the WISKI Server setup.

14.4 Security Monitor

Introduction

The Security Monitor is a small Quarks-based service contained in the **WISKI-*-security-distribution.tar.gz** WISKI Server Manager (WSM) package. It is available at earliest with WISKI 7.4.15 SR6 SP1. It is only installed if JProc is also available - without JProc dependency, the Security Monitor is omitted.

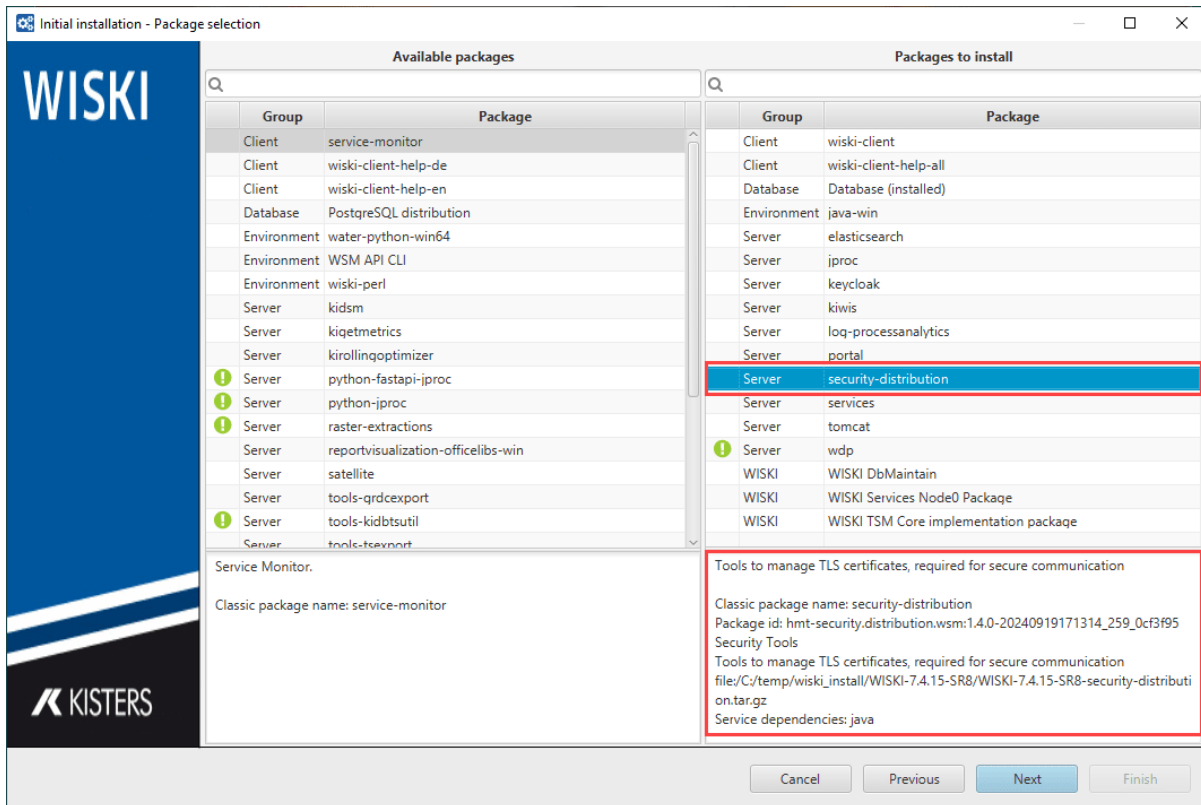
If installed, the service monitors automatically check all certificate chains available in the WISKI Server KeyStore. By default, it checks the validity of the chains every 12 hours (default value). This means that as soon as a certificate is about to expire - by default the threshold is 30 days - an alert is written to the log file in **wiski/services/security-monitor/log/security-monitor.log**.

In the near future, the Security Monitor will also display the certificate expiration status in the JProc client, otherwise known as JProcConsole. This feature is currently under development.

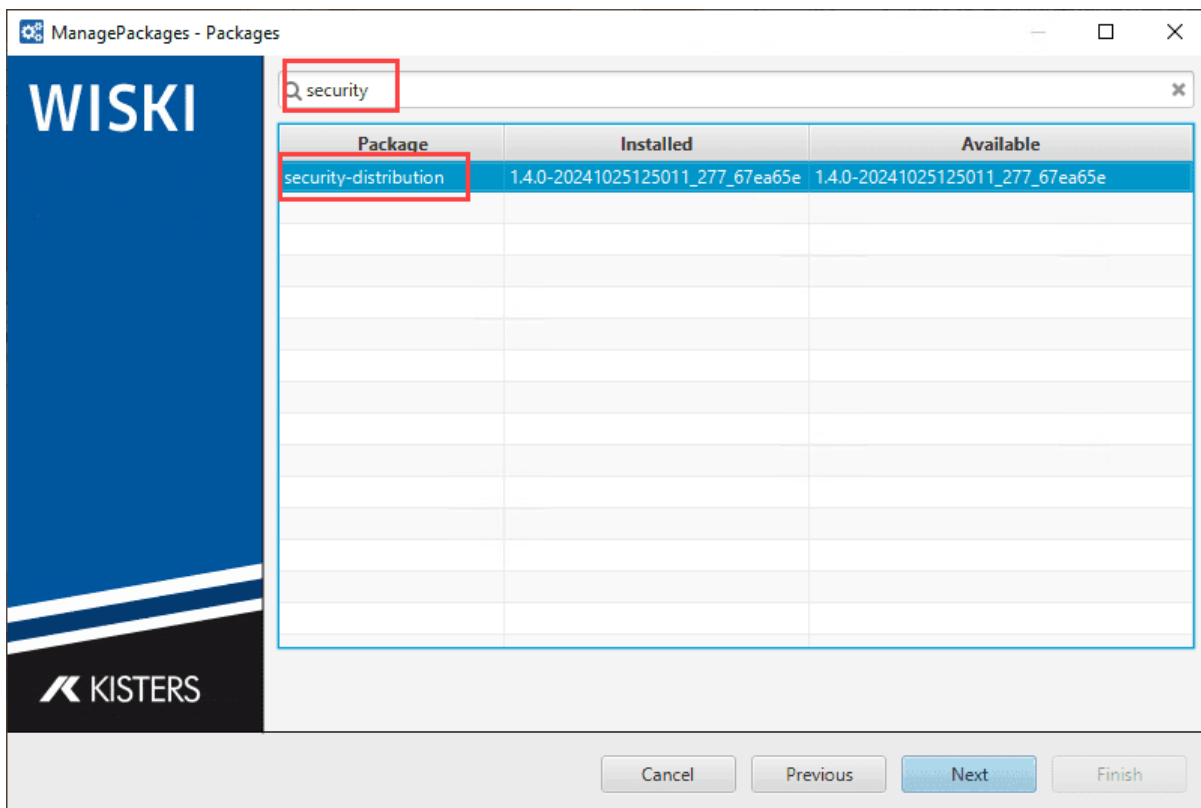
Installation

To install the Security Monitor, you must proceed as follows:

- **Initial Installation or Update:** If you perform an initial installation or update via the WISKI Server Manager, you must select the **security-distribution** package in the corresponding wizard in the **MANAGE PACKAGES - PACKAGES** view and configure it in a later installation step as described below. Further information on installation and updating via the WISKI Server Manager can be found in chapters [Initial Installation Using the WSM \(Empty Database\)](#)^[61] and [Updating an Installation Using the WSM \(Existing Database\)](#)^[76].

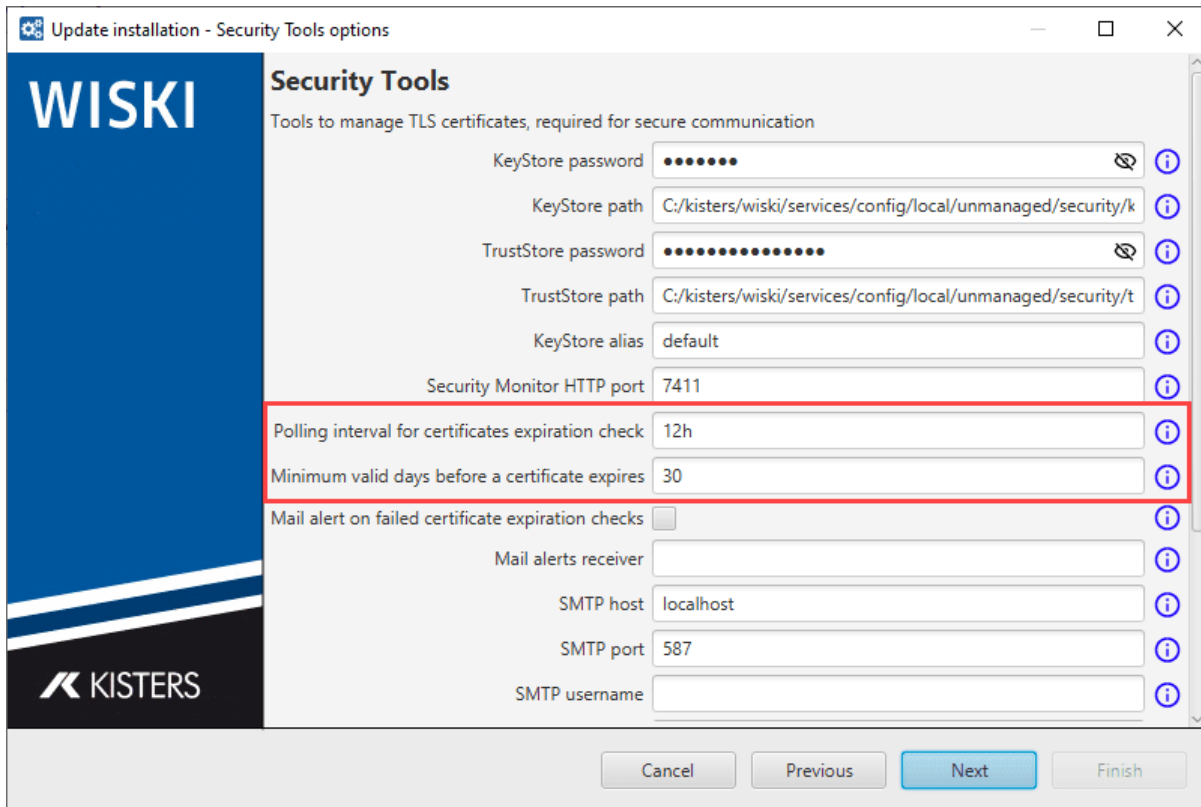


- **Package Management:** If you want to install the Security Monitor subsequently in an existing WISKI Server environment, you must use the Package Management wizard in WISKI Server Manager. In the package selection, search for the package labelled **security-distribution**. Then configure the Security Monitor as described here.



Configuration

The default configuration of the Security Monitor can be changed by installing the package **security-distribution** in WSM Expert Mode. During the installation/update, you can define customised settings for the Security Monitor in the **SECURITY TOOLS OPTIONS** wizard step (see screenshot below). Among other things, you can set the interval at which the validity check of the certificates should take place and how many days before a certificate expires a warning should be issued.



WISKI

Security Tools
Tools to manage TLS certificates, required for secure communication

KeyStore password: [masked] ⓘ

KeyStore path: C:/kisters/wiski/services/config/local/unmanaged/security/k ⓘ

TrustStore password: [masked] ⓘ

TrustStore path: C:/kisters/wiski/services/config/local/unmanaged/security/t ⓘ

KeyStore alias: default ⓘ

Security Monitor HTTP port: 7411 ⓘ

Polling interval for certificates expiration check: 12h ⓘ

Minimum valid days before a certificate expires: 30 ⓘ

Mail alert on failed certificate expiration checks: ☐ ⓘ

Mail alerts receiver: [empty] ⓘ

SMTP host: localhost ⓘ

SMTP port: 587 ⓘ

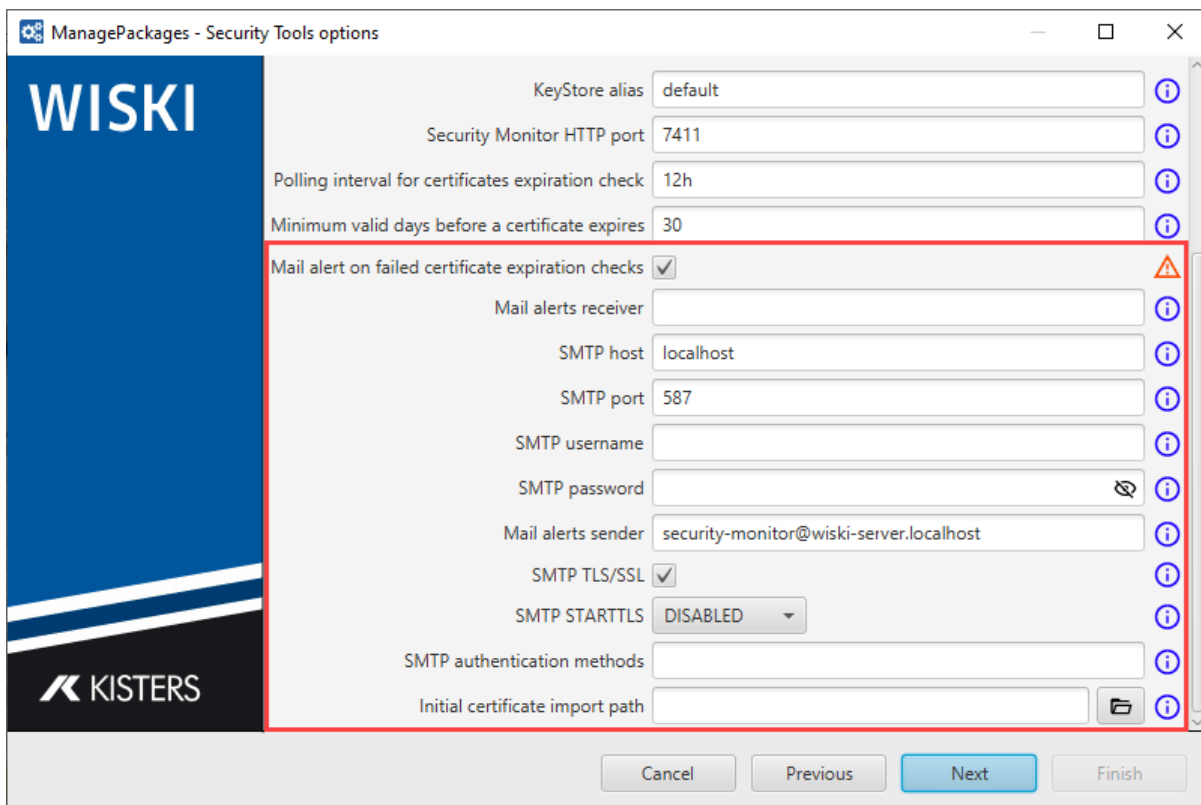
SMTP username: [empty] ⓘ

Buttons: Cancel, Previous, Next, Finish

Note: After installation, the settings can currently only be changed manually in the configuration file at `wiski/services/security-monitor/config/application.properties`. The configuration file also contains documentation on all settings.

Optional e-mail alerts

Optionally, it is also possible to configure e-mail alerts for certificates that exceed the expiration threshold. To do this, you will need to switch WSM Package Management to Expert Mode to see all the options:



WISKI

ManagePackages - Security Tools options

KeyStore alias: default ⓘ

Security Monitor HTTP port: 7411 ⓘ

Polling interval for certificates expiration check: 12h ⓘ

Minimum valid days before a certificate expires: 30 ⓘ

Mail alert on failed certificate expiration checks: ☒ ⓘ ⚠

Mail alerts receiver: [empty] ⓘ

SMTP host: localhost ⓘ

SMTP port: 587 ⓘ

SMTP username: [empty] ⓘ

SMTP password: [masked] ⓘ

Mail alerts sender: security-monitor@wiski-server.localhost ⓘ

SMTP TLS/SSL: ☒ ⓘ

SMTP STARTTLS: DISABLED ⓘ

SMTP authentication methods: [empty] ⓘ

Initial certificate import path: [empty] ⓘ

Buttons: Cancel, Previous, Next, Finish

Check the **MAIL ALERT ON FAILED CERTIFICATE EXPIRATION CHECKS** box and configure the appropriate settings for your SMTP server. The SMTP server must be provided by the customer and is required to send e-mail.

It sends alert e-mails - every 12 hours by default - when the expiration threshold - 30 days by default - is reached. It sends an error e-mail when the certificates expire.

Note that the SMTP server typically requires TLS and therefore its root CA certificate must be imported into the WISKI Server TrustStore. Currently this is also true when using an official root CA, but this limitation will be removed in the future.

15 Security

The Security part of the deployment guide deals with the following topics:

- [Authentication](#)^[131]
- [WISKI Server Security Tool](#)^[154]

15.1 Authentication

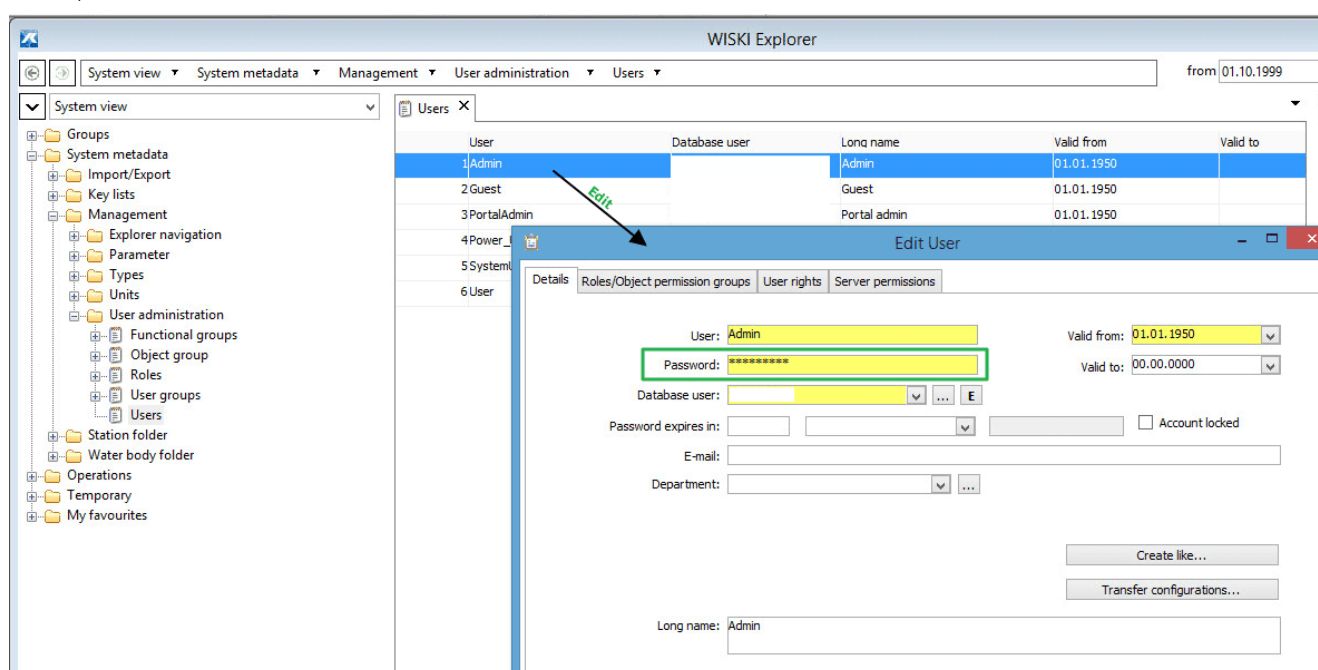
The Authentication part of the deployment guide deals with the following topics:

- [Change Password for a WISKI Client User](#)^[131]
- [Configuring LDAP Authentication](#)^[131]

15.1.1 Change Password for a WISKI Client User

The (default) password for a WISKI7 client user can be changed as follows.

- Login to WISKI via WSM action for Desktop Client starting 'WISKI'.
- Use old current password for client user to log in.
- Open WISKI Explorer (**VIEW: SYSTEM VIEW**, Tree path: **SYSTEM METADATA > MANAGEMENT > USER ADMINISTRATION > USERS**) and edit the user.



- Type in the new password.
- Save and log out.

Now you should log in with the new user credentials.

Note: Be aware that currently there is no check for a similarity of the old and the new password. Also there is no verification of typing errors of the new password (no repeat password dialog). Be aware that momentarily there are no security requirements for the password. If you got any doubts concerning this topic contact your system administrator.

15.1.2 Configuring LDAP Authentication

WISKI7 supports the OpenID Connect standard. Up to version 7.4.5 SR5, only the internal MITREid ID provider was used, using the TCA database as user backend, i.e. the user federation. From version 7.4.5 SR6 onwards, WISKI7 also supports user authentication with external directory services based on OAuth2, to connect an ID provider which supports LDAP (e.g. Keycloak).

The following describe the old and the new system. You will learn how to configure the WISKI server to use an external ID provider that supports external LDAP user repositories (Active Directory, etc.).

Note:

- With WISKI 7.4.13 SR12 Keycloak finally became the default, still supporting LDAP as optional authentication backend. MITREid is deprecated and will be completely removed with 7.4.13 SR13. See the Version Highlight W7.4.13 SR11 WS-7217 KeycloakAsDefault to be obtained from the KISTERS
- From WISKI version 7.4.5 onwards, the old dispatcher LDAP authentication plugin is deprecated and was removed with version 7.4.5 SR6.

Configuring LDAP authentication comprises these areas of functionality:

- [Authentication via WISKI LDAP Plugin \(since WISKI 7.4, old authentication\)](#)^[132]
- [WISKI Open ID Connect Configuration \(Keycloak\)](#)^[134]

15.1.2.1 Authentication via WISKI LDAP Plugin (since WISKI 7.4, old authentication)

Note: This way of user authentication is only valid for versions before 7.4.5 SR6. For later versions, only the [WISKI OpenID Connect solution](#)^[134] is supported.

File System

Finally, the WISKI LDAP Authentication Plugin must be installed and configured:

- Open an MS-DOS-Box and navigate to `~/wiski/services`.
- Install the LDAP Authentication Bundle by adapting the XML Cluster Node Profile.
- Edit `WISKIServer_ClusterProfile.xml` and add the `wiski-auth-ldap` component to the dispatcher `dcsInstanceType`:

```
<component name="wiski-auth-ldap" moduleDir="wiski-server" />
```

- Call `perl createOrUpdateDcsServer.pl` to regenerate the cluster node configuration.
- Configure the LDAP Server URL (see also default configuration file) with:

```
notepad dcservice\server_tsmdispatcher\configuration\dcsmgr-wiski-  
auth-ldap-custom.properties
```

- For tests in the local KISTERS network in Aachen, the following connection string should be used (adapt it to your credentials):

```
ldap://CN=Cedric Jonas,OU=Development,OU=Users  
Aachen,DC=kisters,DC=de/<your_windows_domain_password>@kis-dc-  
01.kisters.de:389
```

- Add this URL with key 'ldap.url' to the configuration file (see also default configuration file):

```
ldap.url=ldap://CN=Cedric Jonas,OU=Development,OU=Users  
Aachen,DC=kisters,DC=de/<your_windows_domain_password>@kis-dc-  
01.kisters.de:389
```

- If this does not work, check your connection string via Windows permissions dialogs.
- Restart the KiTSM Dispatcher in order to perform the next steps.

KiTSM Webinterface

- Connect to the KiTSM server administration under `http://tsmservice:7420/TsmSAdmin`.
- Go to tab page **AUTHORISATION**.
- Go to tab page **DISPATCHER**.
- Go to tab page **PASSWORD SERVER CONNECT**.
- Click on **[+]** in the heading line.
- Specify:

Name:	WISKI7 LDAP
Plug-in Java Class:	de.kisters.tsmsystem.dispatcher.server.AuthDispatchHandlerOSGi (select from drop down list box)

Parameter:	WISKI7-Auth-LDAP (click on the proposed item below)
Status:	On
Cache Time:	0

Home TSM Cluster Log Viewer Tenants **Authorisation** Preferences

Dispatcher Tenants and Permissions User-Admin

Password Server Connect Server Map Logging

Edit Password Server Connection

Create / Edit Password Server Connection

Name: WISKI7 LDAP

Plug-in Java Class: de.kisters.tsmssystem.dispatcher.server.AuthDispatchHandlerOSGi

Parameter: WISKI7-Auth-LDAP
Currently AuthDispatchHandlerOSGi services with following parameters are available: WISKI7-Auth, WISKI7-Auth-LDAP

Status: On

Save Reset

KISTERS install user Cluster Admin KiAdminDomain 07.11.2014 14:00:24 CET

A Server Map entry must be added to insert the WISKI LDAP authentication plugin into the authentication chain:

- Connect to the KITSM server administration under <http://tsmserver:7420/TsmSAdmin>.
- Go to tab page **AUTHORISATION**.
- Go to tab page **DISPATCHER**.
- Go to tab page **SERVER MAP**.
- Click on [+] in the heading line.
- Specify:

Name:	1002 (one larger as the largest one on the SERVER MAP tab page)
Index:	On
Status:	Yes
Try next rule in case of no permission:	No
Expression:	true
Password Server:	WISKI7 LDAP

Home TSM Cluster Log Viewer Tenants Authorisation Preferences

Dispatcher Tenants and Permissions User-Admin

Password Server Connect Server Map Logging

Change / Create Server Rule

Change / Create Server Rule

Index: 1002

Status: On

Try next rule in case of no permission: Yes

Allow foreign tenant Authentication: No

Expression: true

Password Server: WISKI7 LDAP

Save Reset

KISTERS install user Cluster Admin kiAdminDomain 07.11.2014 14:40:06 CET

Home TSM Cluster Log Viewer Tenants Authorisation Preferences

Dispatcher Tenants and Permissions User-Admin

Password Server Connect Server Map Logging

	Key	Index	Rule Status	Server Status	Continue	Expression	Password Server	Multi Tenants
	1001	1000	On	On	Yes	callingClient == "TsmAdmin" callingClient == "tca"	Tsm Password Service	No
	2001	1001	On	On	Yes	true	WISKI7	No
	3001	1002	On	On	Yes	true	WISKI7 LDAP	No

KISTERS install user Cluster Admin kiAdminDomain 07.11.2014 14:48:32 CET

WISKI7 Client Configuration

- Add your users to the WISKI7 User Administration, use this syntax:
`<USER NAME>@<TENANT>`
 with the user name being the same as the operating system login name.
 How to check your OS user name: With the **whoami/upn** command, you can check your LDAP user name that you have to create in the WISKI user administration.
- Finally, edit the file **WISKI.ini** and add the following sections and keys:

```
[LoginDialog]
TsmLDAPLogin=1
tsmenableallsettings=0

[TSM]
tenant=WISKI
host=tsmsserver
port=7430
```

The **WISKI.exe** tries to connect to the KiTSM Server and sends the name of the user that is currently logged in. If the login fails, the login dialog is opened with user name and password greyed out. Use this chance to correct tenant, host and port.

15.1.2.2 WISKI Open ID Connect Configuration (Keycloak)

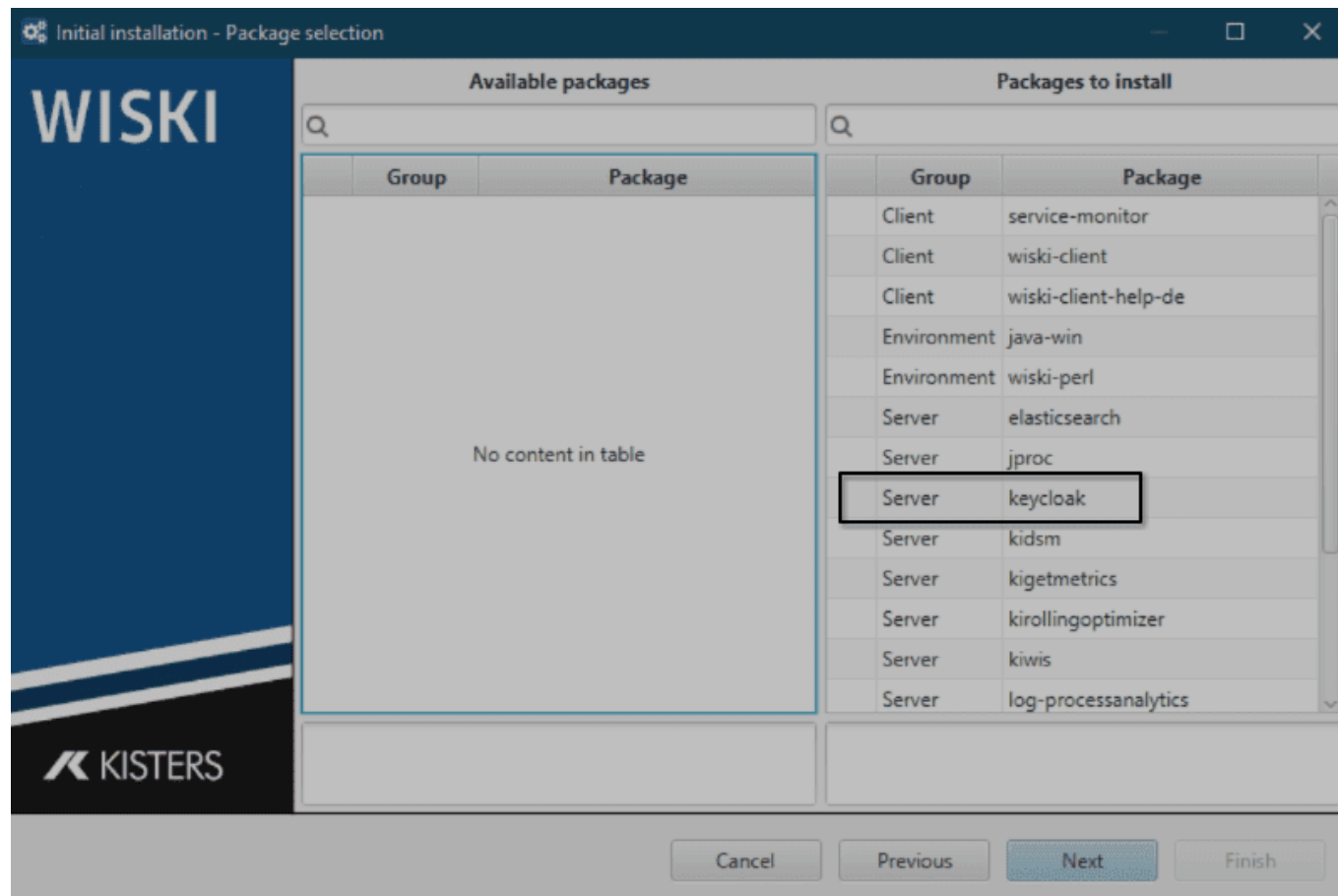
WISKI7 Server uses the OpenID Connect standard for user authentication. The default Identity (ID) Provider delivered with WISKI is Keycloak. Keycloak provides optional support for external authentication backends like e.g. LDAP (including Active Directory).

The next subsections explain how to set up and configure Keycloak. Also the necessary WISKI7 client and server configuration steps are explained.

Generally you have to perform the steps depicted below.

Installing Keycloak

Keycloak is mandatory and installed by default by the WISKI Server Manager (see screenshot below).

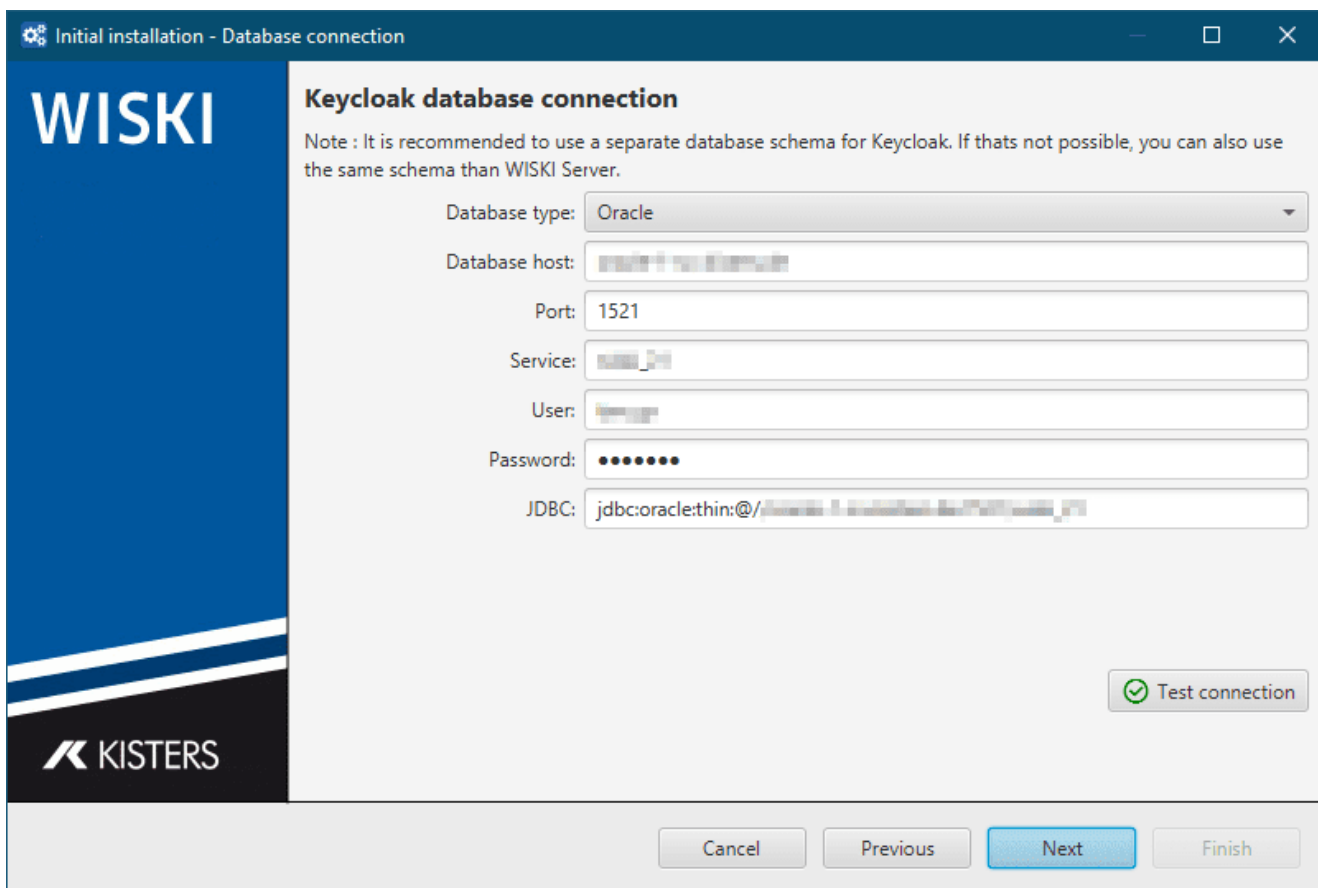


In the next steps of the installation or update process, you will have to enter your credentials for the Keycloak database connection.

Note: We recommend using your own database schema for Keycloak. It is possible to install Keycloak inside the WISKI schema, but we do not recommend this for security reasons.

Proceed like this:

- Before the **[NEXT]** button is activated, the connection must be tested. Click **[TEST CONNECTION]** for this.



WISKI

Keycloak database connection

Note : It is recommended to use a separate database schema for Keycloak. If thats not possible, you can also use the same schema than WISKI Server.

Database type:

Database host:

Port:

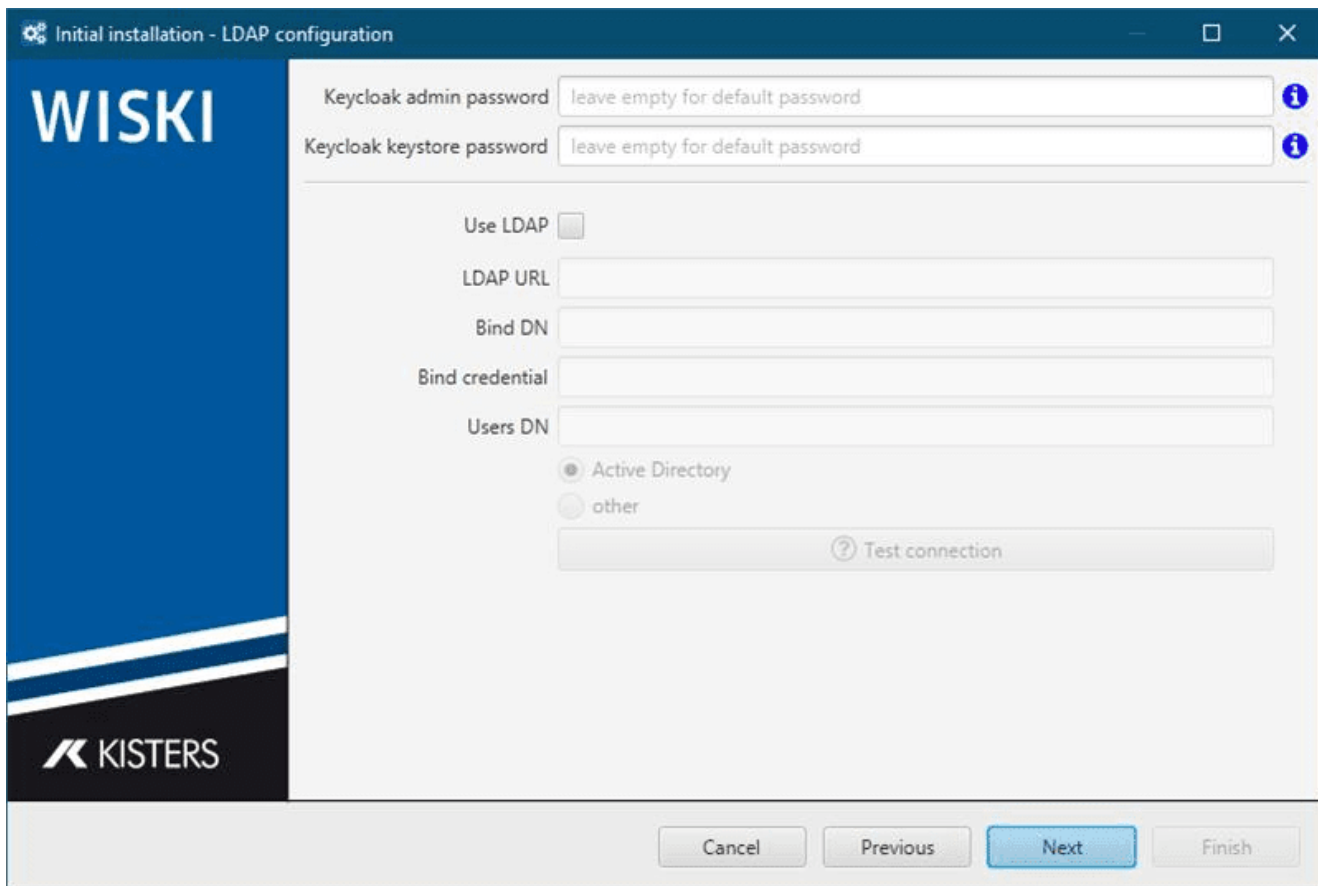
Service:

User:

Password:

JDBC:

- The next window asks for the Keycloak admin password. Note that the Keycloak admin user is named **keycloak** and its default password is **12years**.
- Optionally, the configuration of the LDAP backend can be enabled by clicking **USE LDAP**.
- Fill in your LDAP credentials.



WISKI

Keycloak admin password ⓘ

Keycloak keystore password ⓘ

Use LDAP ☐

LDAP URL

Bind DN

Bind credential

Users DN

☒ Active Directory
☐ other

More information on this step can be found [here](#).

- Finally the connection to the LDAP server can be tested.

Running Keycloak by JProcController

Starting and stopping Keycloak is handled by the JProcController and can be monitored in the WSM Service Monitor.

Web Consoles

You can address two types of consoles which are explained below.

Admin Console

The Keycloak Web Admin Console is available at **https://localhost:7415/**.

The name of the Keycloak admin user is **keycloak**.

Note: The Keycloak admin user is **not** the WISKI admin user. Keycloak has it's own admin user!

The default password of the Keycloak admin user is **12years**.

Note: It is not possible to log in as a normal user here.

User Console

The Keycloak Web User Console is available at **https://localhost:7415/admin/kisters-water-default/console** if the default realm (**kisters-water-default**) is being used.

Each Keycloak or LDAP user is allowed to log in here. Use it to e.g. check the validity of the LDAP configuration without involving WISKI Server.

Note: It is not possible to login with the Keycloak admin user here.

Configuring Keycloak for LDAP User Department Attribute ("department")

When you install Keycloak with the WISKI Server Manager, it is already pre-configured: A **realm** is created and configured, **multiple clients** are created and configured as well as a **user federation** and a **default admin user** for Keycloak. Also the database connection is correctly set. All corresponding properties files are adapted according to your settings during installation.

In some cases, it may be required to modify the default Keycloak configuration. These cases are explained in the next sub sections.

Scope Protocol Mapper

Additional attributes like the user department can be read from LDAP and passed to the system if required. To do so, the LDAP user federation in Keycloak provides mappers. These mappers will map LDAP directory attributes to the Keycloak internal user model.

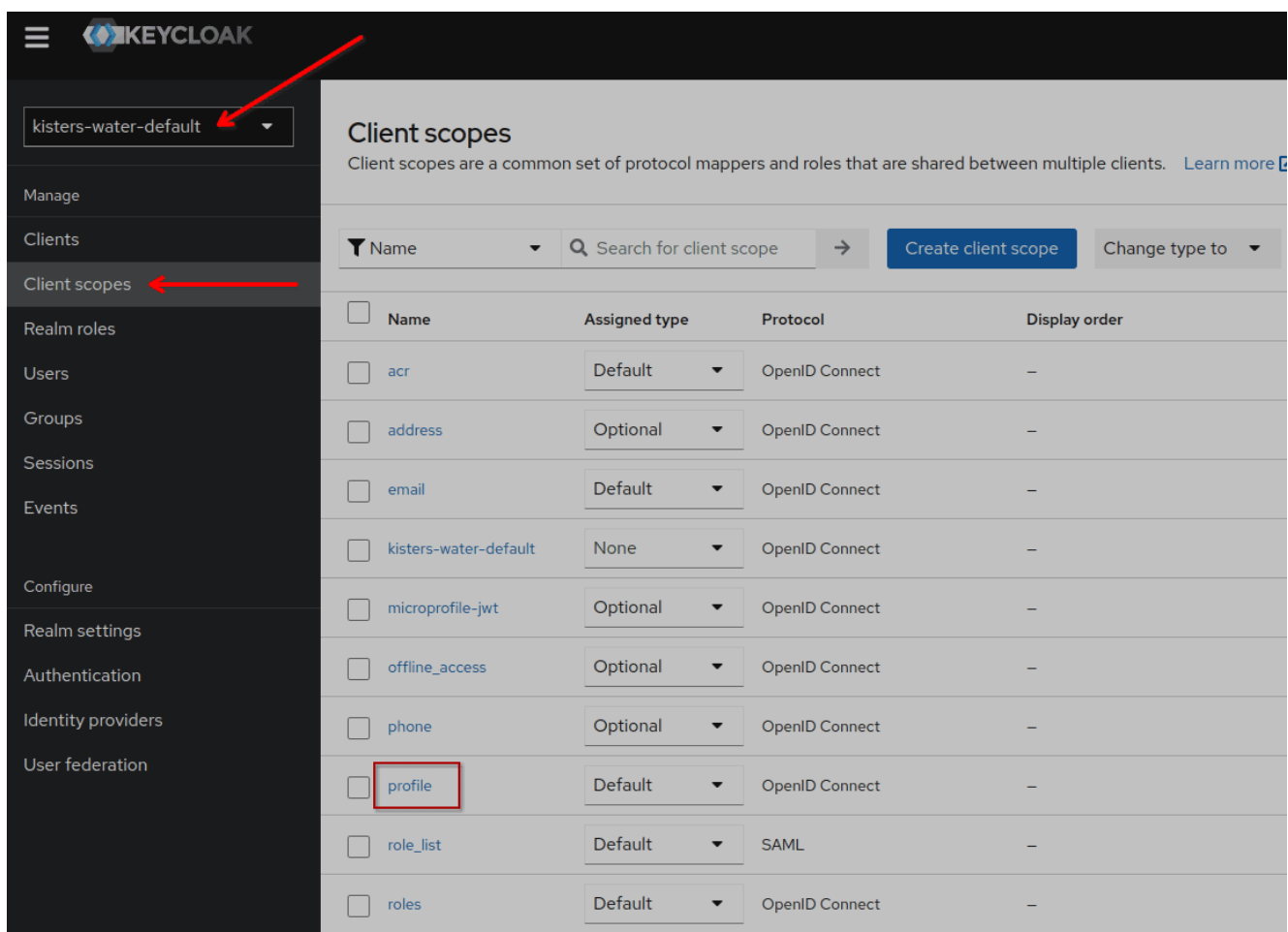
When WISKI users log in, the WISKI Server can request additional user information like the department. For this, it needs to know the key of the additional user information. This is configured in **wiski/services/dcserver/server_tsmdispatcher/configuration/dcsmgr-wiski-auth-oidc-default.properties** by default:

```
# User info mapping: username, long name & e-mail are standardized (https://openid.net/specs/openid-connect-core-1_0.html#StandardClaims)
# and will be replicated automatically from the UserInfo endpoint of the OpenID Connect ID Provider if available there.
# Following non-standardized items can be made available via the specified key value at the UserInfo endpoint of the ID Provider.
authentication.oidc.mapping.department=department
```

Note: This configuration must typically not be changed.

The ID Provider - Keycloak - will provide this additional information in the **department** field thanks to a so called "scope protocol mapper". This means that if the WISKI Server requests additional user information with e.g. the scope **profile**, the field department will be filled with help of all mappers configured for the **profile** scope. In our specific case, the scope protocol mapper for the department will take the department information from the Keycloak user model mentioned above and publish it with the **department** key.

- To create and configure this mapper, log in to the Keycloak Web Admin Console.
- Select the **kisters-water-default** realm and switch to the **CLIENTS SCOPES** menu item on the left:

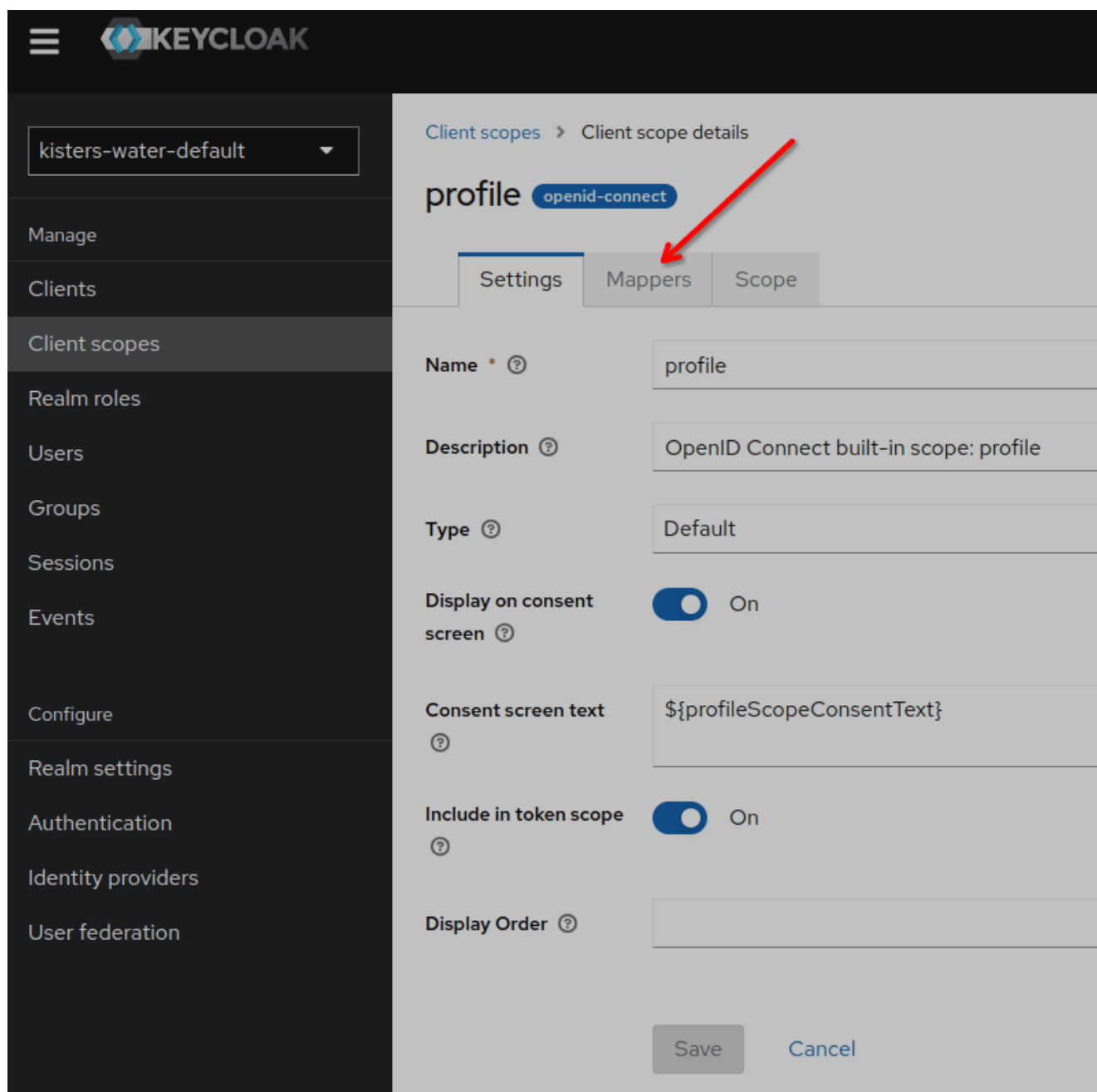


Client scopes
Client scopes are a common set of protocol mappers and roles that are shared between multiple clients. [Learn more](#)

Filter: Name | Search for client scope | Create client scope | Change type to

☐	Name	Assigned type	Protocol	Display order
☐	acr	Default	OpenID Connect	–
☐	address	Optional	OpenID Connect	–
☐	email	Default	OpenID Connect	–
☐	kisters-water-default	None	OpenID Connect	–
☐	microprofile-jwt	Optional	OpenID Connect	–
☐	offline_access	Optional	OpenID Connect	–
☐	phone	Optional	OpenID Connect	–
☐	profile	Default	OpenID Connect	–
☐	role_list	Default	SAML	–
☐	roles	Default	OpenID Connect	–

- Then click on the **profile** scope and select the mappers:



The screenshot shows the Keycloak Admin Console interface. On the left is a dark sidebar with a menu. The top of the sidebar has the Keycloak logo and a hamburger menu icon. Below that is a dropdown menu currently showing 'kisters-water-default'. The menu items include: Manage, Clients, Client scopes (highlighted), Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is light gray. At the top, it shows 'Client scopes > Client scope details'. Below this, the title 'profile' is followed by a blue button labeled 'openid-connect'. There are three tabs: 'Settings' (active), 'Mappers' (indicated by a red arrow), and 'Scope'. The 'Mappers' tab is currently empty. The 'Settings' tab contains several fields: 'Name' (required, with a help icon) set to 'profile'; 'Description' (with a help icon) set to 'OpenID Connect built-in scope: profile'; 'Type' (with a help icon) set to 'Default'; 'Display on consent screen' (with a help icon) set to 'On' via a toggle switch; 'Consent screen text' (with a help icon) set to '\${profileScopeConsentText}'; 'Include in token scope' (with a help icon) set to 'On' via a toggle switch; and 'Display Order' (with a help icon) which is empty. At the bottom right of the settings are 'Save' and 'Cancel' buttons.

- Next, select **ADD MAPPER** and use the context menu item **BY CONFIGURATION**:

The screenshot shows the Keycloak Admin Console interface. On the left is a dark sidebar with navigation links: Manage, Clients, Client scopes (selected), Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is titled 'Client scopes > Client scope details' and shows the 'profile' scope with the 'openid-connect' protocol. Below this are tabs for 'Settings', 'Mappers' (selected), and 'Scope'. A search bar 'Search for mapper' is present. An 'Add mapper' button is highlighted with a red arrow. Its dropdown menu is open, showing two options: 'From predefined mappers' and 'By configuration', with the latter highlighted by another red arrow. Below the menu is a table of existing mappers.

Name	Mapper Type
given name	
middle name	Token mapper
birthdate	Token mapper
zoneinfo	Token mapper
updated at	Token mapper
profile	Token mapper
username	Token mapper
locale	Token mapper
picture	Token mapper
family name	Token mapper

- In the upcoming screen, select **USER ATTRIBUTE** as mapper type.
You will then be forwarded to a page for the configuration of the new mapper:

Client scopes > Client scope details > Mapper details

Add mapper

If you want more fine-grain control, you can create protocol mapper on this client

Mapper type: User Attribute

Name *: Department

User Attribute: department

Token Claim Name: department

Claim JSON Type: String

Add to ID token: Off

Add to access token: Off

Add to userinfo: **On**

Multivalued: Off

Aggregate attribute values: Off

Save Cancel

Note:

- Make sure that the mapper type is of type **USER ATTRIBUTE**.
 - The name could be "Department" or anything other meaningful name since it is only a display name.
 - The user attribute is read from the internal Keycloak user model and should be **department**.
 - The Token Claim Name must be **department** and corresponds to the value configured by default in **dcsmgr-wiski-auth-oidc-default.properties** as mentioned above.
 - The JSON Type must be **string**.
 - ADD TO USERINFO** must be the only enabled switch.
- Finally, save the new scope protocol mapper.
- Keycloak is now able to pass over the department information from the Keycloak internal user model to the WISKI Server if requested.

User Federation Mapper

- The next step is to configure another mapper on the **LDAP User Federation level**.
Note: This is the most important part as it must be customised since each customer LDAP directory structure is different, and each customer must decide about what the **department** field in his directory tree is.
- To do so, log in again to the Keycloak Web Admin Console, select the **kisters-water-default** realm and switch to the **USER FEDERATION** menu item on the left:

The screenshot shows the Keycloak administration interface. On the left is a dark sidebar with a menu. The 'kisters-water-default' realm is selected at the top. The 'User federation' menu item is highlighted at the bottom. The main content area is titled 'User federation' and includes a description: 'User federation provides access to external databases and directories, such as LDAP and Active Directory.' Below this are two buttons: 'Add new provider' and 'Manage priorities'. A card for 'KISTERS WATER LDAP' is displayed, showing it is 'Enabled'.

KEYCLOAK

kisters-water-default

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

User federation

User federation provides access to external databases and directories, such as LDAP and Active Directory. [Learn more](#)

Add new provider Manage priorities

KISTERS WATER LDAP

Ldap Enabled

- Click on the **KISTERS WATER LDAP** User Federation and select the **MAPPERS** tab on the top.
- Then click **ADD MAPPER**:

The screenshot shows the Keycloak administration console. On the left, the 'User federation' menu item is selected. The main panel displays the 'LDAP' configuration for 'Mappers'. The 'Mappers' tab is active, and the 'Add mapper' button is highlighted with a red arrow. Below the search bar, a list of LDAP attributes is shown:

Name
creation date
email
first name
last name
modify date
username

- Again, choose a meaningful display name for the mapper, e.g. "Department".
- Make sure that the mapper type is of type **USER-ATTRIBUTE-LDAP-MAPPER**. The internal user model attribute is **department**, as we configured in the scope protocol mapper above.

Note: The LDAP attribute itself is "department" in our example but must be customised for each installation. This attribute name must match to your LDAP directory tree structure. Read-only must be the only enabled switch.

KEYCLOAK

kisters-water-default

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

User federation > Settings > Mapper details

Create new mapper

Name * ⓘ Department

Mapper type * ⓘ user-attribute-ldap-mapper

User Model Attribute ⓘ department

LDAP Attribute ⓘ department

Read Only ⓘ ☒ On

Always Read Value From LDAP ⓘ ☐ Off

Is Mandatory In LDAP ⓘ ☐ Off

Attribute default value ⓘ

Force a Default Value ⓘ ☐ Off

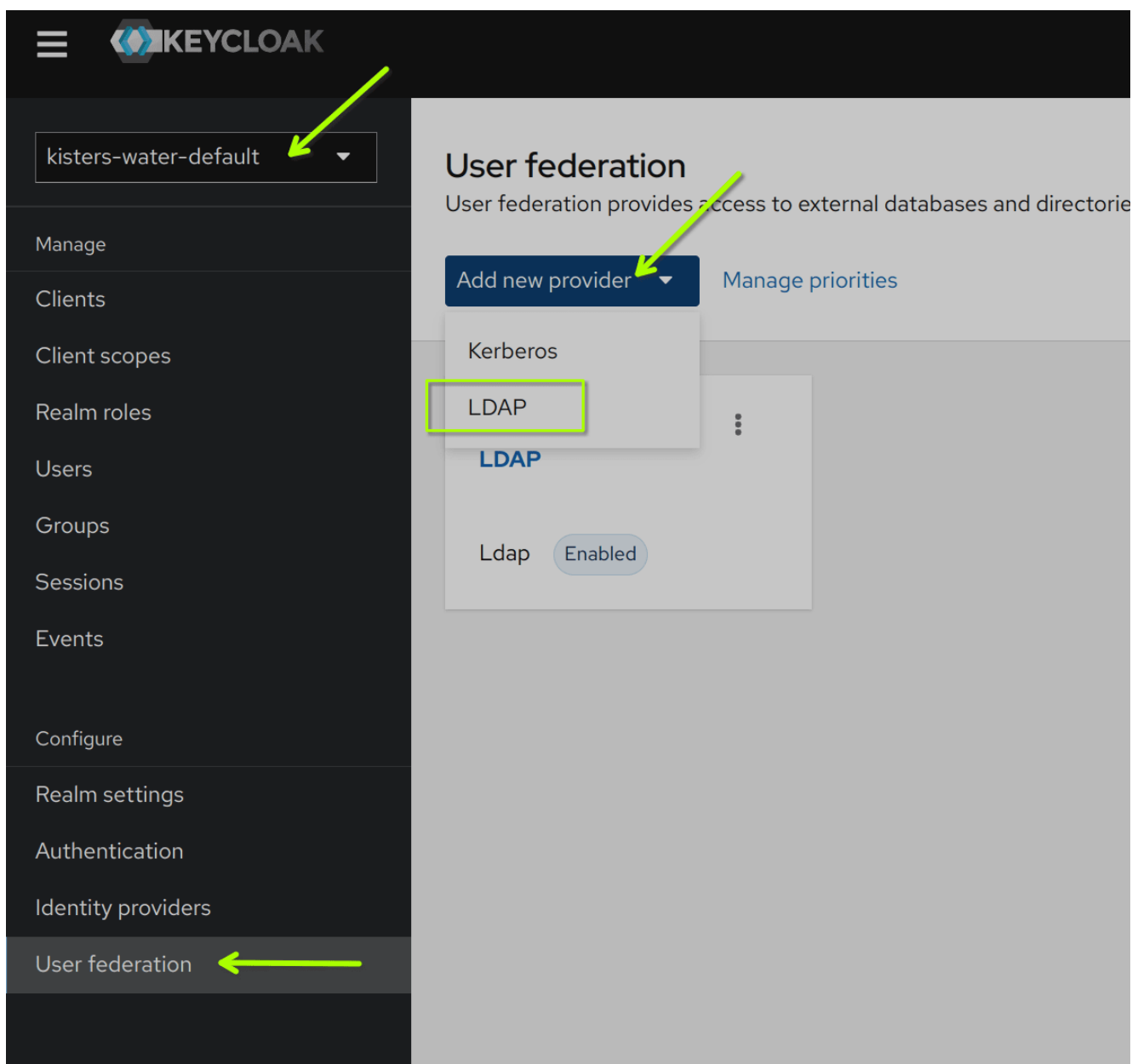
Is Binary Attribute ⓘ ☐ Off

- Finally, save the new mapper on the LDAP User Federation level.
Each user logging in to WISKI for the first time from now on will get its department configured. But make sure that the department is also available in the WISKI database, otherwise it will not be assigned.

Add LDAP User Federation Provider

In some situations it can be useful to configure more than one User Federation Provider, for example, if you need to configure more than one users DN or if you have multiple LDAP backends (e.g. a second backend for backup purposes).

- To do so, login to the Keycloak Web Admin Console.
- Select the **kisters-water-default** realm.
- In the left menu, select **USER FEDERATION** and click on **ADD NEW PROVIDER** and select **LDAP** as provider.



You are redirected to the **ADD LDAP PROVIDER** page where you first configure the **general options** first.

User federation > Add LDAP provider

Add LDAP provider

General options

UI display name * ⓘ my-ldap-user-federation

Vendor * ⓘ Other

- Define a meaningful UI display name; this defines how the provider is displayed in the Keycloak tabs and dialogs.
- As **vendor**, choose **ACTIVE DIRECTORY** for Active Directory implementations or **OTHER** for any other LDAP directory implementation.
- Scroll down to the section **CONNECTION AND AUTHENTICATION SETTINGS**:

Connection and authentication settings

Connection URL * ?

Enable StartTLS ? ☐ Off

Use Truststore SPI ?

Connection pooling ? ☐ Off

Connection timeout ?

Bind type * ?

Bind DN * ?

Bind credentials * ?

- Set the **connection URL**, this is the URL to your LDAP server (the machine where the LDAP server is running).
- In case you want to use a secure connection, use **ldaps://** and the port provided by the customer or IT responsible. Keep in mind that if the LDAP server uses a certificate signed by a non-official root CA, the root CA must be added to the WISKI Server TrustStore as currently Keycloak cannot consume the Windows TrustStore.
- Click on **TEST CONNECTION** to check if you can connect to the configured server. This will only test if connectivity is given on the network layer: check if the machine is reachable and - optionally - if a TLS connection can be established.
- Configure the setting **BIND DN**, the distinguished name of the connection parameters to LDAP including the necessary rights, consisting of common name (CN), OU and DC. It is comparable to the username to login to the LDAP server and must be provided by the customer / IT responsible.
- Configure the **bind credential**, i.e. the password of the LDAP user.
- It is comparable to the password to login to the LDAP server and must be provided by the customer / IT responsible.
- Click on **TEST AUTHENTICATION** to check if you can authenticate to the configured server with the provided credentials. For this, the bind DN settings for LDAP access are checked (whether LDAP accepts the user).
- Scroll down to section **LDAP SEARCHING AND UPDATING**:

LDAP searching and updating

Edit mode * ⓘ READ_ONLY

Users DN * ⓘ OU=dev,DC=wtr,DC=kisters,DC=de

Username LDAP attribute * ⓘ uid

RDN LDAP attribute * ⓘ uid

UUID LDAP attribute * ⓘ entryUUID

User object classes * ⓘ inetOrgPerson, organizationalPerson

User LDAP filter ⓘ

Search scope ⓘ One Level

Read timeout ⓘ

Pagination ⓘ ☐ Off

- Set the **EDIT MODE** entry to **READ_ONLY**.
 - The entry **USERS DN** is the full DN of the LDAP tree where your users are.
This DN is the parent of the LDAP users. It could be for example **ou=users, dc=example, dc=com** assuming that your typical user will have DN like **uid=john, ou=users, dc=example, dc=c [...]**; the distinguished name of the current user on LDAP; this indicates how the users are organised on the LDAP server, with the organisation units (OU) and the domain controller(s) (DC).
This setting must be provided by the customer/IT responsible.
 - The setting for **USERNAME LDAP ATTRIBUTE** can typically be kept as-is and is used to map the username from LDAP.
This name of the LDAP attribute is mapped as Keycloak user name. For many LDAP server vendors it can be **uid**. For Active directory it can be **SAMAccountName** or **cn**. The attribute should be filled for all LDAP user records you want to import from LDAP to Keycloak; default is **cn** when using Active directories, should equal the required Windows user name, see below.
In case of issues, ensure that this attribute contains the expected user name in the LDAP directory.
 - **UUID LDAP ATTRIBUTE**: Name of LDAP attribute, which is used as unique object identifier (UUID) for objects in LDAP. For many LDAP server vendors it's **ENTRYUUID** however some are different. For example for Active directory it should be **OBJECTGUID** If your LDAP server really doesn't support the notion of UUID, you can use any other attribute, which is supposed to be unique among LDAP users in tree. For Example **UID** or **ENTRYDN**.
 - **USER OBJECT CLASSES**: All values of LDAP objectClass attribute for users in LDAP divided by comma. For example: **INETORGPERSO, ORGANIZATIONALPERSON**. Newly created Keycloak users will be written to LDAP with all those object classes and existing LDAP user records are found just if they contain all those object classes.
 - Optionally, you can configure an **additional LDAP filter** for filtering searched users:
 - **USER LDAP FILTER**: Additional LDAP filter for filtering searched users. Leave this empty if you don't need additional filters. Make sure that it starts with (' and ends with ').
 - **SEARCH SCOPE**: For one level (default setting), we search for users just in DNs specified by User DNs.
 - The search scope can be set to **SUBTREE** in case your users are not all located directly in a tree, but organized in sub trees.
The search is then performed for all subtrees.
 - Save the new user federation.
- Note:**
- The number of OU and DC entries depend on the actual LDAP setup.
 - All other settings can stay as is.
 - Ensure that the setting for **USERNAME LDAP ATTRIBUTE** refers to user names that do not contain blanks; WISKI7 does not support user names with blanks!
 - Ensure also that the default value for **USERNAME LDAP ATTRIBUTE** matches the required user name.

Adding Clients

For e.g. usage of the WISKI REST API via OIDC authentication it may be necessary to add new clients.

- To do so, login to the Keycloak Web Admin Console, select the **kisters-water-default** realm.
- Select **CLIENTS** from the navigation menu on the left and click on **CREATE CLIENT**..
- In the first step (**GENERAL SETTINGS**), ensure that the **client type** is set to **OpenID Connect**.

The screenshot shows the Keycloak Web Admin Console interface. On the left, the navigation menu is visible with the 'kisters-water-default' realm selected at the top. The 'Clients' menu item is highlighted. The main content area is titled 'Create client' and shows the 'General Settings' step. The form includes the following fields:

- Client type**: Set to 'OpenID Connect'.
- Client ID**: Set to 'my-rest-client'.
- Name**: Set to 'My REST Client'.
- Description**: Empty text box.
- Always display in UI**: Toggle switch set to 'Off'.

At the bottom of the form, there are three buttons: 'Next' (highlighted), 'Back', and 'Cancel'.

- Select a **client ID** which has to be used later for authentication in your own application.
- Select a meaningful **display name**.
- Optionally, a **description** can also be added.
- Press **NEXT**.

1 General Settings

2 **Capability config**

3 Login settings

Client authentication ⓘ ☒ On

Authorization ⓘ ☐ Off

Authentication flow

☒ Standard flow ⓘ

☐ Implicit flow ⓘ

☐ OAuth 2.0 Device Authorization Grant ⓘ

☐ OIDC CIBA Grant ⓘ

☒ Direct access grants ⓘ

☒ Service accounts roles ⓘ

Next Back Cancel

- In the next step (**CAPABILITY CONFIG**), ensure that **CLIENT AUTHENTICATION** is enabled.
- **STANDARD FLOW** must be disabled while **DIRECT ACCESS GRANTS** and **SERVICE ACCOUNTS ROLES** must be enabled.
- Press **NEXT**.

1 General Settings

2 Capability config

3 Login settings

Root URL ?

Home URL ?

Valid redirect URIs ?

+ Add valid redirect URIs

Valid post logout redirect URIs ?

+ Add valid post logout redirect URIs

Web origins ?

+ Add web origins

Save Back Cancel

- This step (**LOGIN SETTINGS**) can be left as-is. Click **SAVE**.
- To get the credential - client secret - of your new client, switch to the **CREDENTIALS** tab and make your secret visible.

Clients > Client details

my-rest-client OpenID Connect

Clients are applications and services that can request authentication of a user.

Settings Keys **Credentials** Roles Client scopes Service accounts roles Sessions Advanced

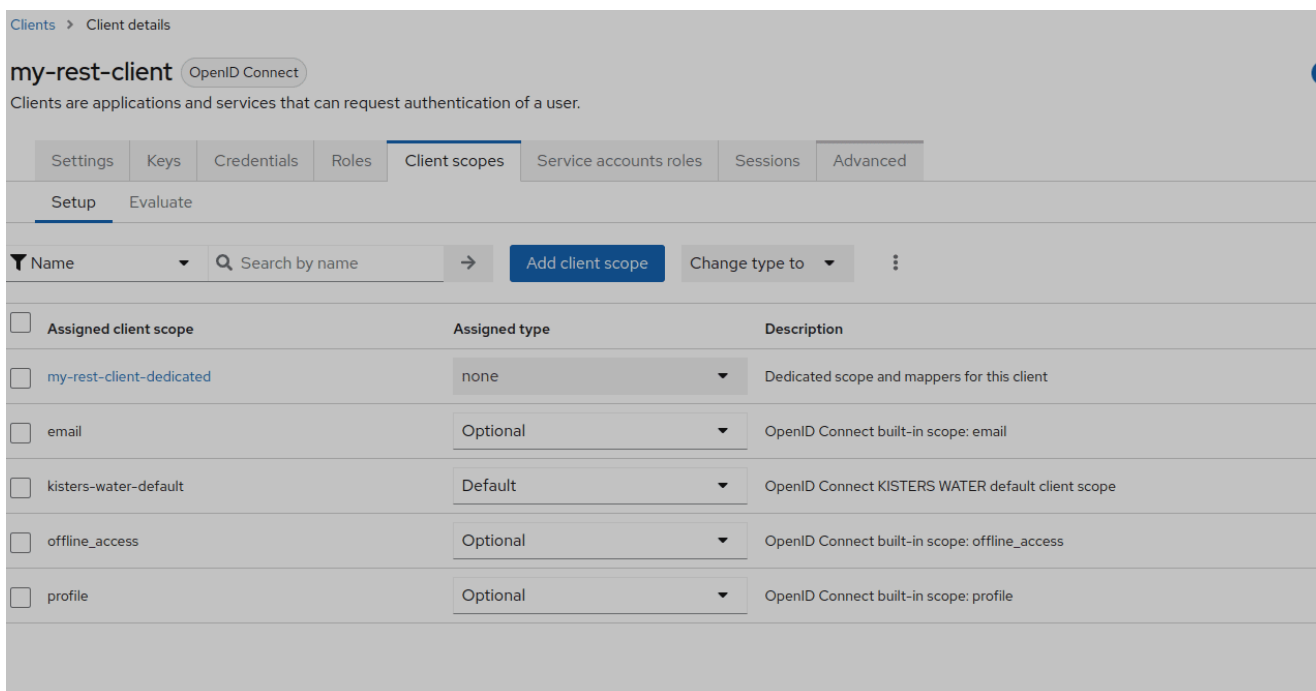
Client Authenticator ? Client Id and Secret

Save

Client secret ppTRZcLMqBGizDTAShu2bv0f9MR5Otw8 Regenerate

Registration access token ? Regenerate

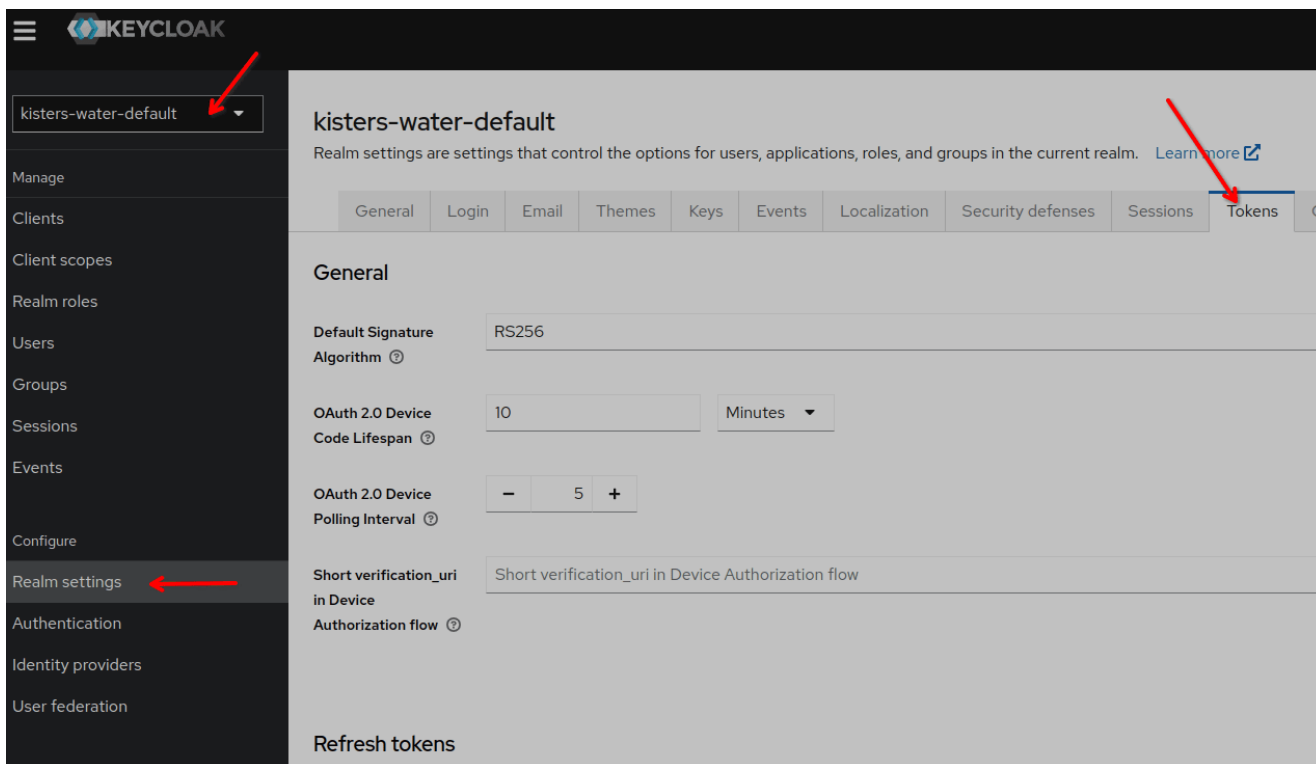
- As a last step, switch to the **CLIENT SCOPES** tab:



- Ensure that the configured client scopes correspond to what is displayed in the screen above.
- Remove everything which is not displayed in the screenshot above, add **kisters-water-default** and double check if the assigned type of each scope is valid (**OPTIONAL** or **DEFAULT** - as displayed in the screenshot). The client can now be used!

Changing "Remember Login" Session Lifetime

- Login to the Keycloak Web Admin Console, switch to the **kisters-water-default** realm and select the **REALM SETTINGS** menu item.



- In the realm settings, select the **TOKENS** tab and scroll down to the **ACCESS TOKENS** section:

Access tokens

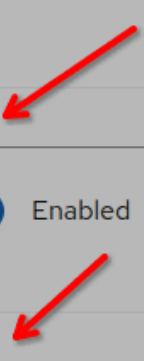
Access Token Lifespan [?]	5	Minutes ▼
It is recommended for this value to be shorter than the SSO session idle timeout: 30 minutes		
Access Token Lifespan For Implicit Flow [?]	15	Minutes ▼
Client Login Timeout [?]	1	Minutes ▼
Offline Session Max [?]	90	Days ▼



- The **OFFLINE SESSION MAX** value is set by default to 90 days. This means that users using the "Remember Login" functionality when logging in to WISKI will be forced to re-authenticate after 90 days. This value can be changed to meet your requirements.
- Switch now to the **SESSIONS** tab and configure the same amount of time in the section **OFFLINE SESSION SETTINGS**:

Offline session settings

Offline Session Idle [?]	90	Days ▼
Offline Session Max Limited [?]	☒ Enabled	
Offline Session Max [?]	90	Days ▼

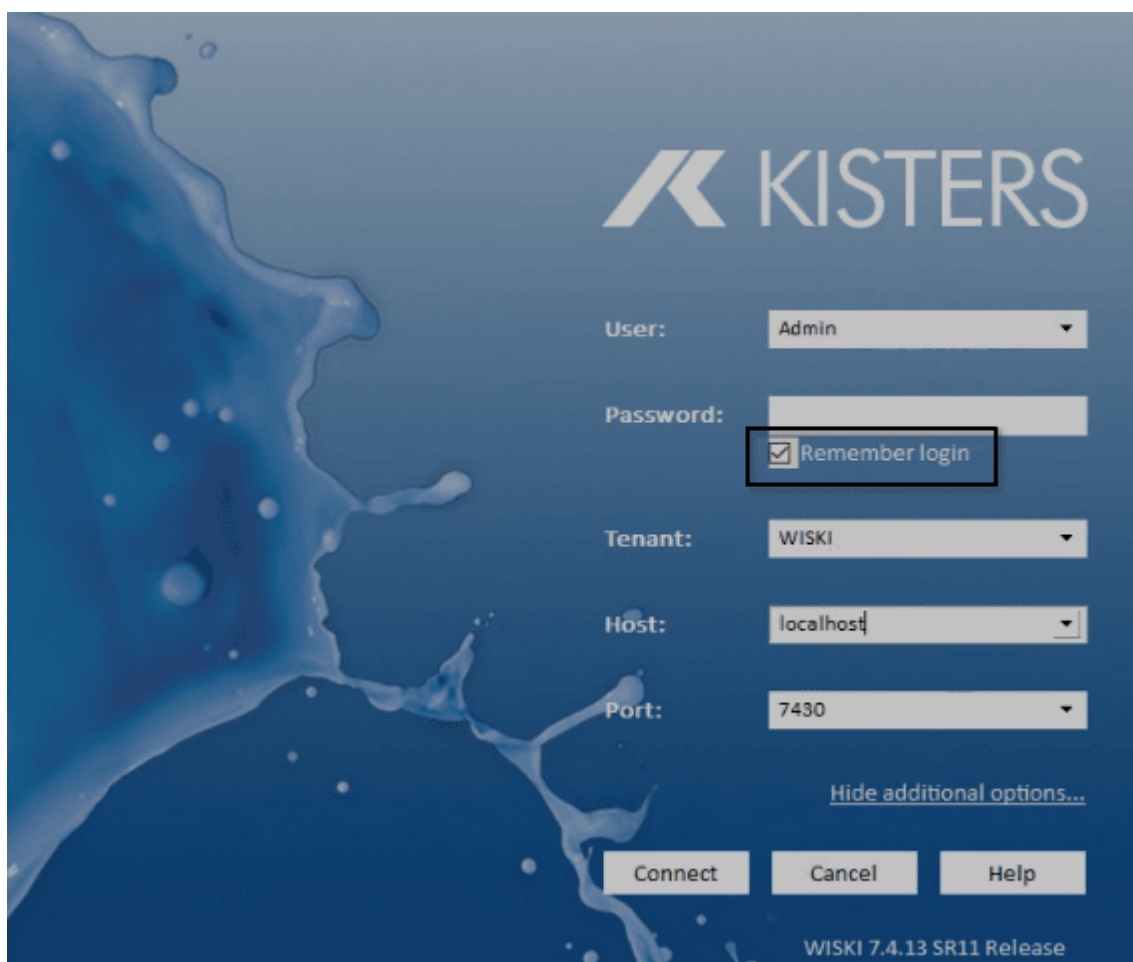


WISKI Desktop Configuration to Remember Logins (Optional)

You can configure WISKI7 in a way that users do not need to always enter user name and password when starting WISKI7. For this, the refresh token provided by the ID provider is stored encrypted. The client remembers the information of the token and provides access to WISKI7 without the login dialog.

The configuration on client side includes the following steps:

- Login as **Admin** user and select **REMEMBER LOGIN**.

A screenshot of the KISTERS login dialog box. The background is a dark blue abstract splash. The KISTERS logo is at the top right. The form contains fields for User (Admin), Password (empty), Tenant (WISKI), Host (localhost), and Port (7430). A checkbox labeled 'Remember login' is checked and highlighted with a black rectangle. Below the fields is a link 'Hide additional options...'. At the bottom are 'Connect', 'Cancel', and 'Help' buttons. The version 'WISKI 7.4.13 SR11 Release' is at the bottom right.

KISTERS

User: Admin

Password:

☒ Remember login

Tenant: WISKI

Host: localhost

Port: 7430

[Hide additional options...](#)

Connect Cancel Help

WISKI 7.4.13 SR11 Release

- Alternatively, set the system setting **REMEMBER LOGIN** under the **LOCAL** node, **LOGIN SETTINGS** sub node to **YES** (see screenshot below).

Filter: remember

System
Local
Login settings
Remember Login

Remember Login

Remember user authentication. Permits login without login dialog. Login dialog can be enabled again with the menu entry "Logout and exit" of main menu "File".

Default value:
No

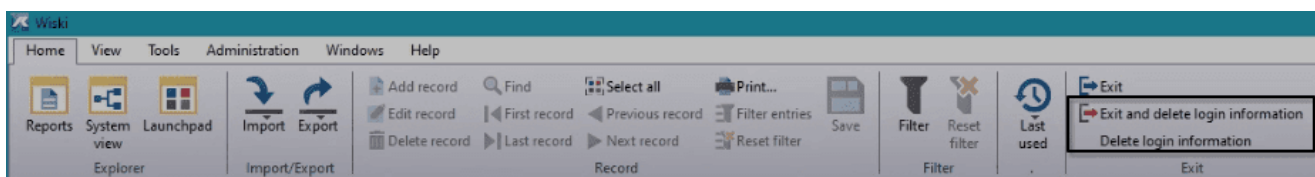
Value for all users:
Yes

Will be overwritten for ulli by:

☐ Show key client / logindialog / rememberlogin Apply

Note: This can also be overwritten for all users.

- Log in with your Windows user credentials.
- Then a refresh token is written to the **wiski_prof_<user>.ini**, the token is AES encrypted.
- When you then exit WISKI7, your next program start will take the refresh token into account; consequently, no login dialog is presented.
- To delete the refresh token, exit WISKI7 with the ribbon menu entry **HOME > EXIT AND DELETE LOGIN INFORMATION** (see screenshot below).



- After the refresh token is deleted, the login dialog appears again for the next start of WISKI7.

15.2 WISKI Server Security Tool

The WISKI Server Security Tool is a small utility delivered with WISKI Server to simplify handling of security related issues like

- password encryption
- HTTPS tests
- KisRPC encryption
- certificates handling
- KeyStores handling
- etc.

The tool is available in **services/wiski-server/bin/security-tool.jar** and must be executed in the working directory **services/wiski-server/bin/**.

Note the available online help for the security tool which should be addressed since information in this topic here could be older or even outdated.

```
cd services/wiski-server/bin/
../../java/bin/java -jar security-tool.jar
```

Executing the steps above will print the - always up-to-date - online help:

```
Usage: <main class> [options] [command] [command options]
Options:
  --help

Commands:
  genDefaultCertsSetup      Generates default WISKI Server KeyStore with
                             standard password and self-signed certificate
  Usage: genDefaultCertsSetup [options]
  Options:
    --help

  buildCertificateChain      Builds - best effort - an ordered certificate
                             chain from multiple source certificates. The resulting chain will
                             be written in the WISKI KeyStore folder inside chain.crt.
  Usage: buildCertificateChain [options]
  Options:
    --help

    --certificates
      Multiple files in PEM format containing one or more certificates.
      File suffix is always '.crt' and must not be given here, always
      located in WISKI KeyStore folder.
      Default: [server, rootCA]
    --chain
      Chain file. File suffix is always '.crt' and must not be given
      here, always located in WISKI KeyStore folder. Default is 'chain'.
      Default: chain
    --extractRootCA
      Extract root Certification Authority Certificate to an own file.
      This is disabled by default. Using the switch enables the
      described behavior, no additional parameter is required.
      Default: false
    --rootCACert
      Root Certification Authority Certificate: suffix is always '.crt'
      and must not be given here, always located in WISKI KeyStore
      folder. Defaults to 'rootCA'.
      Default: rootCA

  encrypt      Encrypt
  Usage: encrypt [options] Plaintext password
  Options:
    --help

    --prompt
      Prompt for plaintext password without echo
  * --type
      Crypto type (userPassword, dbPassword, keyStorePassword,
      keycloakPassword or serviceRunnerPassword)

  test-https      Test HTTPS
  Usage: test-https [options] HTTPS URL
  Options:
    --help

    --trustStoreLocation
      TrustStore location. Defaults to
      "../../config/local/unmanaged/security/truststore.p12"
      Default: ../../config/local/unmanaged/security/truststore.p12
    --trustStorePassword
      TrustStore password. Defaults to "Tpu9oo3HzNRLW5G"
      Default: Tpu9oo3HzNRLW5G
```

```

--trustStoreType
TrustStore type (PKCS12, JKS or windows). Defaults to "PKCS12".
Default: PKCS12

secure-kisrpc      Secure KisRPC by TLS
Usage: secure-kisrpc [options]
Options:
  --help

* --tls
  TLS encryption
  Default: true

createCSR          Create a Certificate Signing Request
Usage: createCSR [options]
Options:
  --help

  --organisation
    Organisation (defaults to KISTERS AG)
  --commonName
    Common Name (defaults to the fully qualified domain name or as
    fallback to the hostname)
  --altDNSNames
    Alternative DNS names: additionally to Common Name and
    comma-separated (defaults to Common Name)

signCSR           Signs a Certificate Signing Request
Usage: signCSR [options]
Options:
  --help

  --csr
    Certificate Signing Request: suffix is always '.csr' and must not
    be given here, always located in WISKI KeyStore folder
    Default: server
  --rootCACert
    Root Certification Authority Certificate: suffix is always '.crt'
    and must not be given here, always located in WISKI KeyStore
    folder
    Default: rootCA
  --rootCAKey
    Root Certification Authority Private Key: suffix is always '.key'
    and must not be given here, always located in WISKI KeyStore
    folder
    Default: rootCA

addCertToKS       Adds a Certificate to the WISKI Server KeyStore
Usage: addCertToKS [options]
Options:
  --help

  --alias
    Alias of the Server Private Key & Certificate to be added to the
    WISKI Server KeyStore
    Default: default
  --serverKey
    Server Private Key: suffix is always '.key' and must not be given
    here, always located in WISKI KeyStore folder
    Default: server
  --certificateChain
    Certificate chain: suffix is always '.crt' and must not be given
    here, always located in WISKI KeyStore folder. Sserver certificate
    including - if available - intermediate certificate(s) and root
    CA. Order is important: first the server certificate, then the
    intermediate certificates in correct order and finally the root
    CA.
    Default: chain

```

```

addCertToTS      Adds a Certificate to the WISKI Server TrustStore
Usage: addCertToTS [options]
Options:
  --help

  --alias
    Alias of the Certificate to be added to the WISKI Server
    TrustStore
    Default: kisters-self-signed
  --certificate
    The Certificate to be added to the WISKI Server TrustStore: suffix
    is always '.crt' and must not be given here, always located in
    WISKI TrustStore folder
    Default: rootCA

genRootCA        Generates a Root Certification Authority Certificate
Usage: genRootCA [options]
Options:
  --help

  --organisation
    Organisation
    Default: KISTERS AG
  --forHost
    For Host information - no functional impact, only adds additional
    information to the certificate name to make clear it's not
    globally valid

```

Available commands

Note:

- Not each command will have a full documentation here. The purpose of the following lines is just to add some additional context to each command, which would simply be too much inside the online help.
- Refer to the online help printing above or - better cause always up-to-date - when calling the tool if you search for detailed documentation for a specific command.

Command	Description
encrypt	<p>This command can be used to encrypt passwords. Ensure to choose the correct password type when doing so.</p> <p>User passwords stored in Keycloak cannot be encrypted by our tool and can - as hashes are stored - not be decrypted.</p>
secure-kisrpc	<p>Allows to enable or disable KisRPC TLS encryption. As it's enabled by default in the meanwhile, this command should not be used any more.</p> <p>Note: It is not recommended to disable KisRPC TLS encryption as you could possibly transmit user data and credentials in clear text via the network.</p>
genDefaultCerts Setup	<p>This command can be used to generate the default certificate / KeyStore setup of WISKI Server. It is also being called by WISKI Server Manager and / or the WISKI Server install / update setup itself when a system is being installed with the default self-signed certificates.</p> <p>Typically, there is no need to call that on your own.</p>
buildCertificateChain	<p>This command is pretty useful and helps to generate a valid server certificate chain. It can consume multiple certificates from multiple files (each file can have 1-n certificates) and will try - on best effort base - to build a valid server certificate chain out of that. This is also being used by WISKI Server Manager.</p> <p>Some restrictions:</p> <ul style="list-style-type: none"> ▪ There must be exactly 1 server certificate

Command	Description
	<ul style="list-style-type: none"> There must be exactly 1 root CA certificate <p>If one or both is missing or available more than once, the tool will fail and inform about the exact issue. Without server and root CA certificates, a chain cannot be built. Also, if there is more than one of them, the tool cannot identify the beginning and the end of the certificate chain.</p> <p>In case of possible intermediate certificates being supplied, the tool will try to build a valid chain and ignore any certificates not required to build the chain.</p> <p>If however you missed to provide some required intermediate certificates the tool will fail and complain about the missing certificate.</p>
genRootCA	Can be used to generate a self-signed root CA private key and certificate. Is also being used by genDefaultCertsSetup (see above).
createCSR	<p>Can be used to generate a server certificate sign request. Is also being used by genDefaultCertsSetup.</p> <p>The sign request - based on a server private key generated at the same time - will be signed by a root CA resulting in a server certificate.</p>
signCSR	Can be used to sign a server certificate sign request with the WISKI Server self-signed root CA. Is also being used by genDefaultCertsSetup .
addCertToKS	Allows to add a server private key and server certificate chain to the WISKI Server KeyStore. Ensure that the alias used here is the one configured in the WISKI Server security configuration (services/config/local/security-default.properties).
addCertToTS	<p>Allows to add a root CA certificate to the WISKI Server TrustStore. The alias is only informational and doesn't affect WISKI Server functionality.</p> <p>Self-signed certificates must be added to the TrustStore if they are not provided by the Windows TrustStore.</p> <p>Official certificates are often available in the Java TrustStore or in the Windows TrustStore and must not be added.</p> <p>Nevertheless, if you use WISKI Server Manager, all root CA certificates will be added as we currently can't check if addition is required or not.</p> <p>As a general rule, you make nothing wrong by adding a root CA certificate here although it's already provided in the Windows or Java TrustStore - it just means additional maintenance efforts in case it has to be exchanged.</p> <p>Do not add server or intermediate certificates here - the TrustStore is only intended for root CA certificates!</p>
test-https	<p>This command can be used to test HTTPS connections especially for debug usage. It allows to specify the TrustStore to be used (any PKCS12 TrustStore or the Windows TrustStore) to call a specific URL.</p> <p>For example, if you want to know if your Windows TrustStore is correctly configured to access e.g. the Service Registry or Portal URL, this tool can be used.</p>

16 WISKI Troubleshooting

The Troubleshooting around WISKI7 and KiTSM comprises these areas of interest:

- [TLS Certificate Issues](#)^[159]
- [Installation of Self-Signed Root CA in Local TrustStore](#)^[160]
- [Database Problems](#)^[162]
- [Error message on WISKI startup "Connect to KWebKitObjCtrl failed -3"](#)^[163]
- [Login Problems](#)^[163]
- [System Startup Problems](#)^[165]
- [Monitoring Your System](#)^[166]
- [Monitoring General KiTSM Server State](#)^[168]
- [Monitoring KiTSM Calculation Queues](#)^[170]
- [Solve performance issues on Windows 2022 with RDP connections](#)^[174]

This list (and the topics contents) will grow by the time...

Note: If the suggested solutions do not work, investigate the problem further, for example by looking at the application log files for detailed error descriptions. Finally: Give us a hint, we will try to find a solution and we will add it here in the Troubleshooting documentation topic.

Also note the FAQs at the end of this documentation.

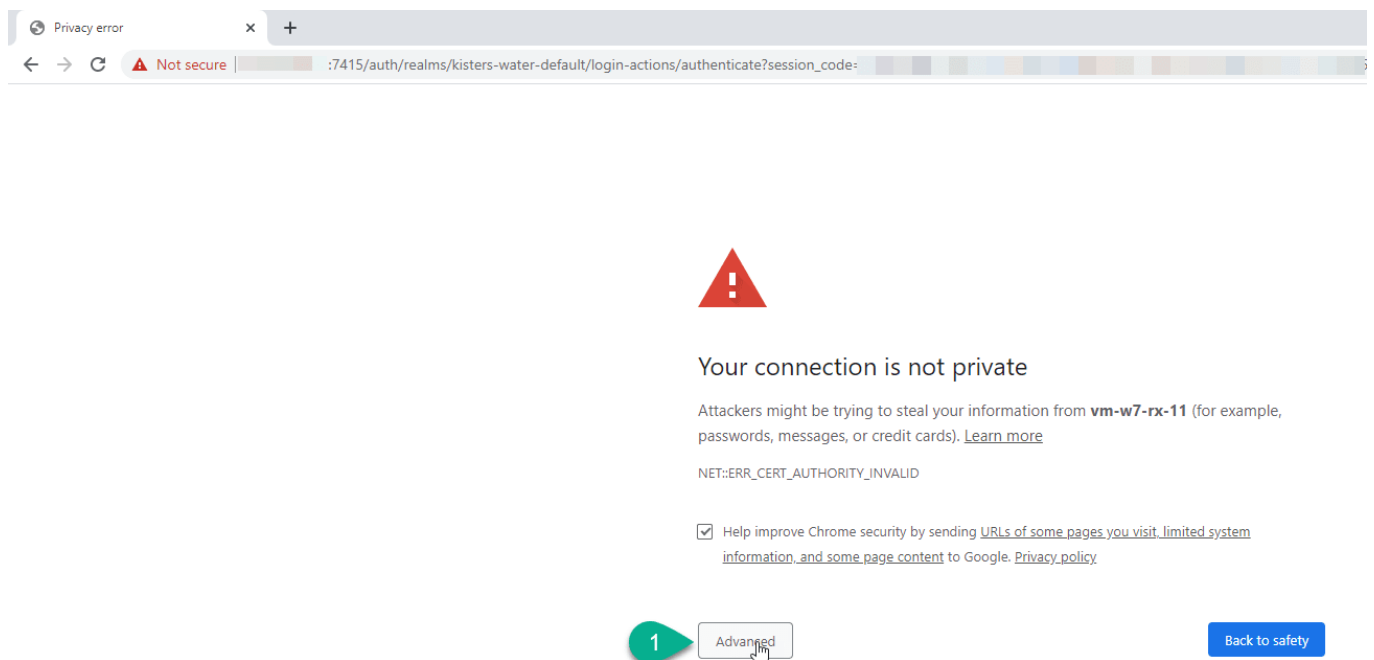
16.1 TLS Certificate Issues

I am using self-signed certificates and cannot access my HTTPS resources any more in my browser

A self-signed certificate is typically signed by an untrusted root certification authority (CA), like the one being generated by the WISKI Server setup (Common Name: **Generated Self-Signed Root CA for WISKI Server <host>**).

Since this certification authority is not known by the system TrustStore (Windows, Linux, Java), the browser cannot trust the remote endpoint providing the self-signed server certificate.

Until around 2022-2023, it was possible to simply add an exception for this certificate & host combination from within the browser:



Since about 2022-2023 however, browsers are becoming more and more strict when it comes to secure communication. So it's often not possible any more to simply add an exception from within the browser. This trend aims at improving end-user security as most users anyway just ignored the message and simply ignored the issue. These changes are not implemented by KISTERS, but by browser vendors.

You could now land on such or similar page without the ability to add an exception. You will get the same message about your connection not being safe, but without a link to proceed to the insecure URL.

To be able to still use such systems with your browser, some workarounds exist:

- Use a certificate provided by yourself as customer.
If you can provide your own certificate, chances are good that they are signed by an official root CA or that the root CA is already installed and trusted on your machines.
If that does not apply, follow the next step.
- Install the WISKI Server's self-signed root Certification Authority in your local operating system TrustStore.
This applies only to WISKI Server installations using generated self-signed root CAs, which is the default setup.
You will need to get the WISKI Server self-signed root CA first.
If your browser is running on the WISKI Server: the WISKI Server self-signed root CA is located in the WISKI Server configuration directory, typically `services/config/local/unmanaged/security/rootCA.crt` or inside the `truststore.p12`.
If your browser is not running on the WISKI Server: the WISKI Server self-signed root CA is accessible via `http://kis-cjonas.kisters.de:7480/security/rootCA.crt`.
If HTTP is not available due to firewall restrictions, use `https://kis-cjonas.kisters.de:7479/security/rootCA.crt` - but then, you will have to be able to define an exception in your browser or use different tools like
`curl -k https://kis-cjonas.kisters.de:7479/security/rootCA.crt > rootCA.crt`
to download the `rootCA.crt`.
In any case, remind to change the hostname to match your system!
If this resource is not available, e.g. on a machine accessing only the Standard Portal, use the following resource:
`https://kis-cjonas.kisters.de:7482/KiWebPortal/rest/wiski7/dataSources/0/wiskiCertificate`
There is also a tool to do that automatically for you if you are using the WISKI Standard Portal, check the WISKI Server Manager package named `water-security.distribution.wsm`.
Once you got access to the `rootCA.crt`, you can add it to the operating system TrustStore. See [Installation of Self-Signed Root CA in Local TrustStore](#) for details on this.
Finally restart your browser, not only one tab or one window but **all processes** and access the TSM Web Admin or the WISKI Standard Portal; no more certificate-related security issue should exist.

I'm installing a WISKI Server satellite and the communication between master & satellite doesn't work

The Eureka log files within DCServer (`services/dcserver/logs/server_*_eureka.log`) and Karaf `node0` (`services/karaf/node0/data/log/karaf_eureka.log`) are a good starting point to check for certificate issues. If there are multiple 10 MB large Eureka log files within a short time range or if the logs are rotating frequently (multiple times per day): all these are hints for probable TLS issues.

Check first the content of these log files.

The problem often is that the whole multi-node system is not using the same root CA certificate. It is theoretically possible to use multiple root CA certificates within one multi-node system, but we do not recommend this since each of these root CA certificates would have to be distributed to the WISKI Server TrustStore or Windows TrustStore for each node. If this is missing, parts of the system will not be able to communicate with other parts. If, for example, the satellite server #1 uses a different root CA certificate than the master node, it cannot trust the master node and Dispatcher and Service Registry connections, among others, will not be able to be established.

Therefore, ensure that the root CA certificate is the **same for all nodes**:

- If your own certificates are used, they are typically the same for all nodes; you can check this in the WISKI Server KeyStore.
- If the default self-signed certificates are used, ensure that the `rootCA.crt` and WISKI Server TrustStore is the **same for all nodes**. Ensure also that each server certificate for the server nodes has been generated on the master node and distributed to the satellite node. Finally, even while being less recommended due to multiple storage of the root CA private key make sure the identical root CA private key is also stored on each satellite.

16.2 Installation of Self-Signed Root CA in Local TrustStore

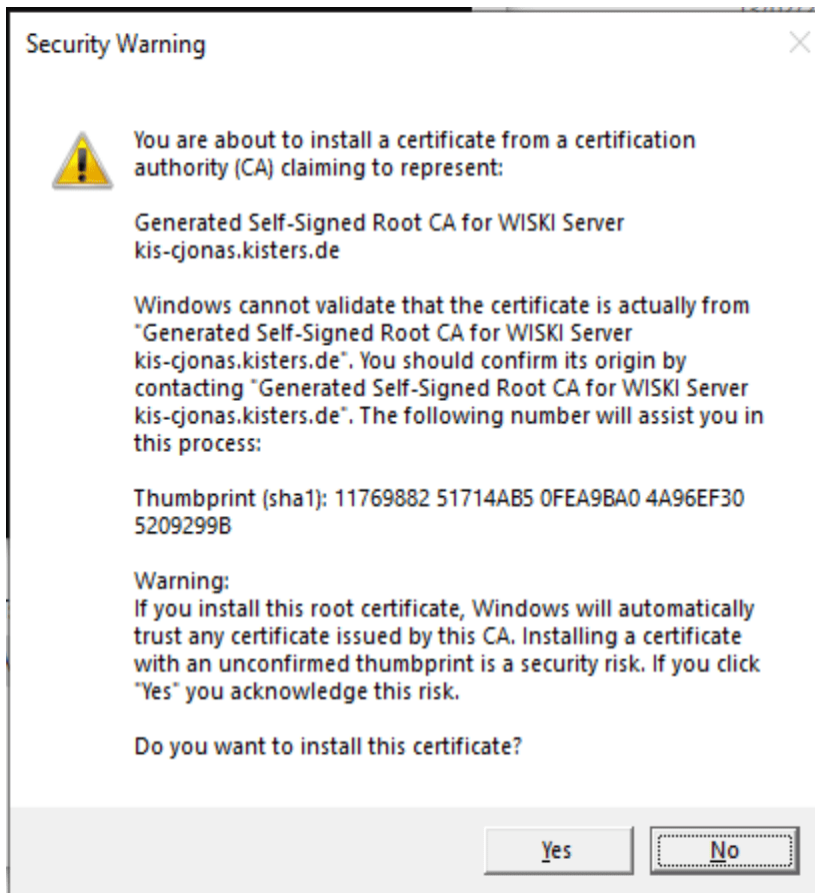
Note: Be careful, since installing a self-signed root Certification Authority into your local system can affect the security of your whole operating system: The operating system (not only WISKI Server) would automatically trust any server providing a certificate signed by this CA! If you feel unsure, stop reading here and ask for support.

In the examples below, `rootCA.crt` is the self-signed root Certification Authority to be installed into the local system TrustStore. The filename could be different.

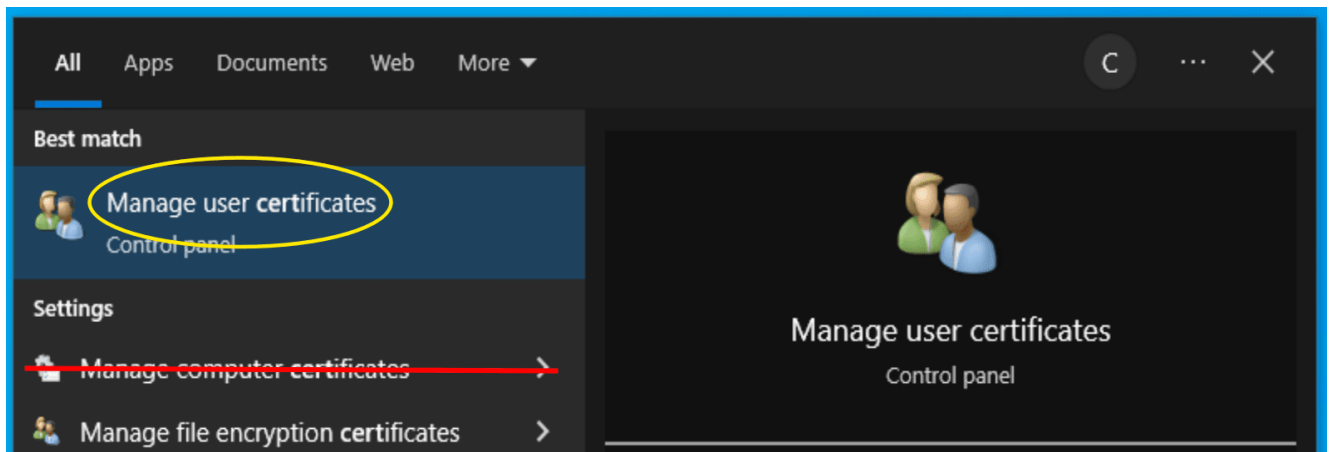
It's recommended to remove the self-signed root CA from the local system TrustStore as soon as it's not needed anymore - see above.

Using Windows (without admin permissions!)

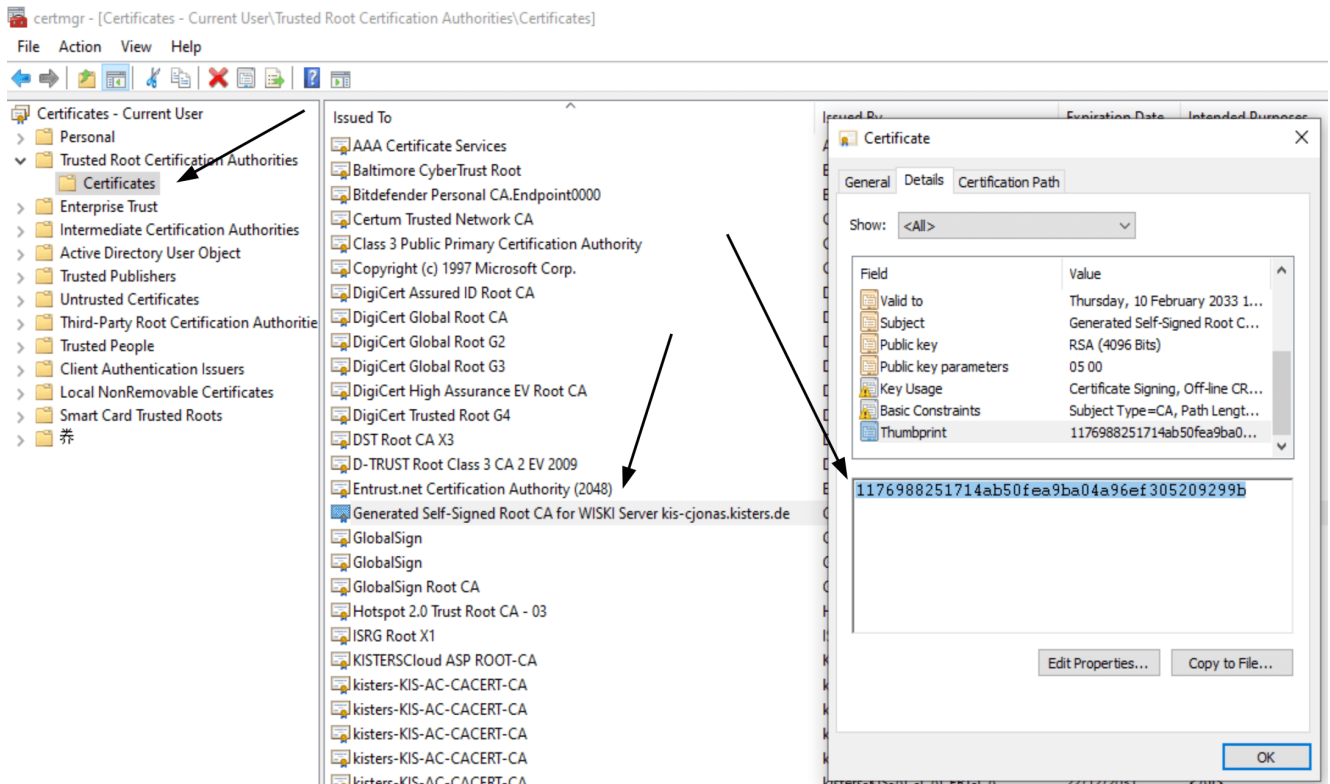
- Execute `certutil.exe -user -addstore root rootCA.crt`.
Windows will then show up a dialog box whose content must be double-checked before confirming.



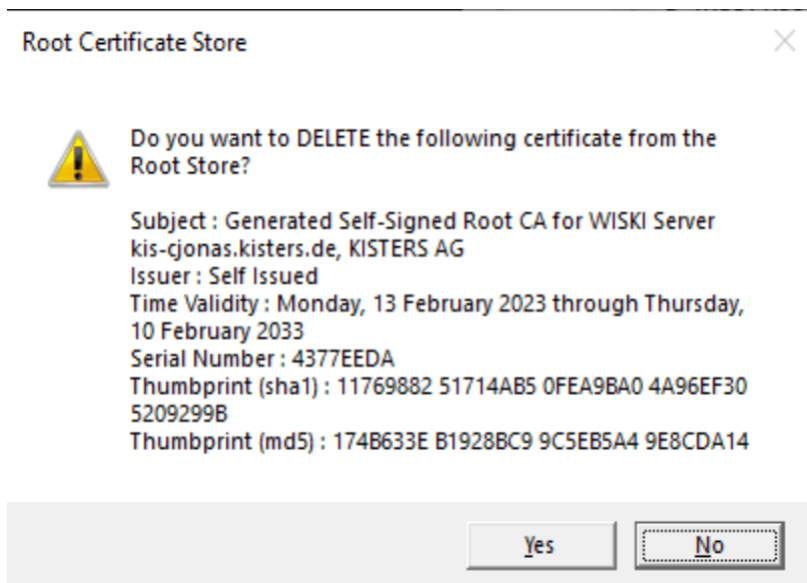
- Use `certutil.exe -user -delstore root <certificate_thumbprint>` to remove it later.
- Check the Windows User Certificate Manager to get the thumbprint.



Note: **Do not use the Windows Computer Certificate Manager!**



- Check the dialog box to ensure you remove the correct certificate before confirming.



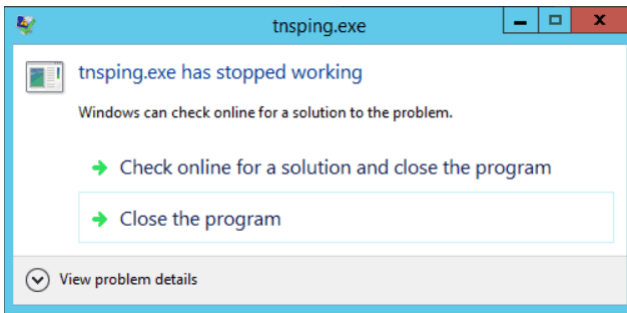
Using Linux

- Execute `sudo trust anchor --store rootCA.crt` to install the CA into the local system TrustStore.
- Use `sudo trust anchor --remove rootCA.crt` to remove it later.
- For information purposes: Use `sudo trust list` to list all installed CA certificates.

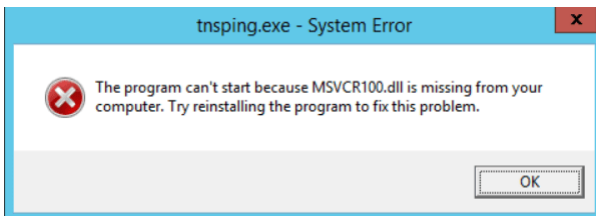
16.3 Database Problems

Error message during database test with ORACLE 12c client.

When you get the following error message when the ORACLE tnsping is executed.



or



The computer misses the **MICROSOFT VISUAL C++ 2010 SP1 REDISTRIBUTABLE PACKAGE (X86)**, [download](#) the package from the Microsoft web page and run the installation.

Other database problems could comprise:

- [Repair Missing Database Constraints](#)^[163]

16.3.1 Repair Missing Database Constraints

If constraints are lost during the database import and export, the following command can repair the database structure.

Solution:

Execute the following command:

```
<root>\services\tsm\bin>tsm-dispatcher.pl database checkschema --repair.
```

16.4 Error message on WISKI startup "Connect to KWebKitObjCtrl failed -3"

The error message **CONNECT TO KWEBKITOBJCTRL FAILED** can be solved by installing the 'Microsoft Media Foundation' feature.

- MS Windows 8 download and install: <https://support.microsoft.com/de-de/help/2929699/media-feature-pack-for-windows-8-1-n-and-windows-8-1-kn-editions-april>
- MS Windows 10 download and install: <https://support.microsoft.com/de-de/help/3010081/media-feature-pack-for-windows-10-n-and-windows-10-kn-editions>

16.5 Login Problems

Note: In case the suggested solutions won't work, investigate the problem further by checking e.g. the application log files for the detailed error descriptions. At the end: give us a hint, we will try to work out a solution and put in here into the Troubleshooting documentation topics.

Login problems could comprise:

- No free TSM state^[164]
- No open tenant MaiDomain or WISKI^[164]
- Login failed for user 'wiskiadm'. while trying to get a pool for tenant 1001^[164]
- TSM/TCA database connection troubleshooting: ODBC connection without DSN^[164]
- TSM/TCA database connection troubleshooting: ODBC connection with DSN on 64-bit computer^[165]
- Database tables missing or not up to date. Ask the administrator to update the database.^[165]

16.5.1 No free TSM state

Possible reason(s):

Maybe the IP address or host name of the computer where the KiTSM is hosted has changed. This leads to two reactions: an additional KiTSM server will be created and the KiTSM Server state is set to **ADM STATE OFF**.

Solution:

- Open the KiTSM System Administration (web interface).
- Use the tab sequence **TSM CLUSTER > TSM SERVER**.
- Check if the TSM server with the correct IP/hostname (usually the first in the list) is set to **ADM STATE ON**, if not start it.
- Note that old KiTSM servers have no more heartbeat but are in **ADM STATE ON** whereas new servers have a heartbeat but are in **ADM STATE OFF**.

Note: When removing old servers, you have to remove the connected background processes as well and you have to create new background processes for the new server. This procedure is explained in detail in Update Dispatcher Database Connection Settings, read from chapter [Check the background process configuration](#) onwards.

Or the Int. usage setting of your KiTSM Server is set to **OFF**, so it can only be used for background calculations.

Solution:

- Open the KiTSM System Administration (web interface).
- Use the tab sequence **TSM CLUSTER > TSM SERVER**.
- Set the Int. usage setting of your KiTSM server to **ON**.

16.5.2 No open tenant MaiDomain or WISKI

Possible reason(s):

The user or password for the connect to the TCA database stored in the database is not correct.

Solution:

Start WISKI7 with the /dbadmin option to verify and possibly reset the database user name and his password (**DATABASE USER** tab). Make sure that you use the database user and password for the dbadmin login, which differs from the WISKI7 login. Check the WISKI7 users on the **USER** tab.

Or the State setting of your domain is set to **INACTIVE**, so that no open tenant can be addressed.

Solution:

See next error message **ERROR WHILE LOGIN INTO...**

16.5.3 Login failed for user 'wiskiadm'. while trying to get a pool for tenant 1001

Possible reason(s):

The TSM Domain configuration has wrong connection settings.

Solution:

1. Open the KiTSM System Administration.
2. Use the tab sequence **TENANTS > TENANTS**.
3. Click the  button.
4. Check the connection parameters and the password in the **EDIT TENANT PARAMETER** tab.

16.5.4 TSM/TCA database connection troubleshooting: ODBC connection without DSN

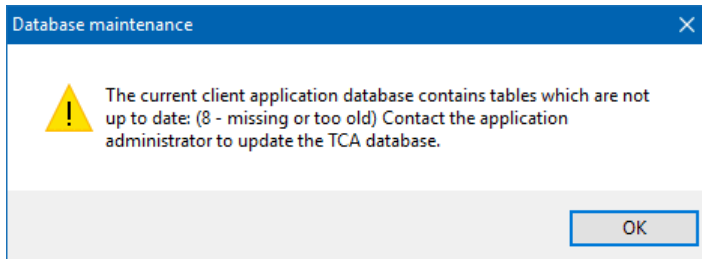
If you want to use an ODBC connection with DSN, try to connect with one of those two possibilities:

- The default DSN if DSN configuration is set to **1** in the ODBCdriver section and its DSN key. Default DSN will be built using this database and server entry: **<database>_on_<server>**.
- The DSN that is currently configured under the ODBCdriver section, key DSN.

16.5.5 TSM/TCA database connection troubleshooting: ODBC connection with DSN on 64-bit computer

If you can not connect because no ODBC is found for the given DSN, display a message box **ODBC DSN NOT FOUND**.

16.5.6 Database tables missing or not up to date. Ask the administrator to update the database.



When you successfully logged in, the above message could come up. It shows that the database of your current client does not match the structures of your client software version. In this case you have to perform **DbMaintain.exe**.

You could also get a message that reads **IN THIS DATABASE TABLES WERE FOUND WHICH ARE NEWER THAN THIS VERSION**. In this case no further action is necessary.

Note: Make sure you started **DbMaintain.exe** from the correct sub folder.

Note: **DbMaintain.exe** can also be called from the command line using the following arguments to perform certain tasks:

- check table structure (argument: **CheckTableStructure**)
- update statistics (argument: **UpdateStatistic**)
- fix table to functional group assignment (argument: **ToFixTable2FuncGroupAssignment**)
- synchronisation of unit groups and units between TSM and TCA (argument: **SyncUnitGroups**)
- synchronisation of parameter types between TSM and TCA (argument: **SyncParameterTypes**)
- synchronisation of basic data between TSM and TCA (argument: **SyncBasicData**)
- update time zone historical information on TCA (argument: **UpdateTZHistInfos**)
- update role atomic permissions on TSM (argument: **UpdateRoleAtomicPermissionsOnTsm**)

Each argument can be used with the parameter **yes** or **no**, e.g.: **/CheckTableStructure yes** or **/ToFixTable2FuncGroupAssignment no**.

16.6 System Startup Problems

Note: There are two database connections in your system, one for the KiTSM database (accessible through the tab sequence **TENANTS > TENANTS** (select your tenant) > **TENANT**: (your tenant), one for the TCA database (accessible through the tab sequence **TENANTS > TENANTS > EXTRA DATABASE**).

All database connection information is stored in the **\$SOFTWARE_INST_DIR\bin\runtime\TSM-Dispatcher.properties** file.

All other database connection information is stored in the database itself. That is the reason why you can not simply move a database from one server to the other. You have to specify all connection settings again on the new database location! See [Duplicating WISKI7/KiTSM database](#) for further information.

Note: Read through the [Updating Dispatcher Database Connections](#)^[94] topic for further information.

System startup problems could comprise:

- Dispatcher DB status is set to "Stop" (red) in the WSM Service Monitor^[166]
- KiTSM Dispatcher status is set to "Stop" (red) in the WSM Service Monitor^[166]
- Webservice status is set to "Stop" (red) in the WSM Service Monitor^[166]
- TSM server status is set to "stop" (red) in the WSM Service Monitor^[166]
- There are only "?" in the WSM Service Monitor instead of status indicators^[166]

16.6.1 Dispatcher DB status is set to “Stop” (red) in the WSM Service Monitor

Possible reason(s):

Login to Dispatcher database is not correct, or the database server connection cannot be established.

Solution:

- Check the settings in the `$SOFTWARE_INST_DIR\bin\runtime\tsm-dispatcher.properties` file.
- Check if the database server is running.

16.6.2 KiTSM Dispatcher status is set to “Stop” (red) in the WSM Service Monitor

Possible reason(s):

TSM dispatcher can not work with dispatcher database.

Solution:

- Check the log files in install dir folder `\wiski\services\tsm\bin\runtime`.

16.6.3 Webservice status is set to “Stop” (red) in the WSM Service Monitor

Possible reason(s):

Tomcat has problems. Usually a port problem.

Solution:

- Check if the Tomcat ports are blocked by another program. Use e.g. `CurrPorts`.

16.6.4 TSM server status is set to “stop” (red) in the WSM Service Monitor

Possible reason(s):

TSM server could not be started.

Solution:

- Check the log files in install dir folder `\wiski\services\tsm\bin\runtime`.

16.6.5 There are only "?" in the WSM Service Monitor instead of status indicators

Possible reason(s):

The service has not started at all.

Solution:

- Start the system manually with `jprocontroller.pl -daemon` or check the KiTSM service status in your local system settings.

16.7 Monitoring Your System

To monitor the status and performance of your KiTSM system, you have quite some possibilities:

- [Using the JProcConsole](#) ¹⁶⁶
- [Using the Web Admin User Interface](#) ¹⁶⁸

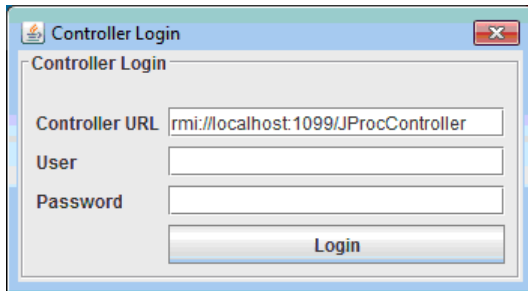
16.7.1 Using the JProcConsole

KiTSM provides a console program named JProcConsole that gives you an overview about the default services included in KiTSM. The JProcConsole can be started from the command prompt.

```
$SERVICES_INST_DIR\jprocontroller\bin\jproconsole.pl
```

-daemon

JProcConsole prompts with a login screen:



User: admin

Password: admin

A successful installation could look like this:



You can manipulate the refresh rate with **EDIT > PREFERENCES** or double-clicking the right corner entry of the status bar.

You can manually refresh the display using the corresponding button or switch the GUI language with the **LANGUAGE** main menu entry.

See also [Troubleshooting](#) if one or more of the entries above are not running (green).

16.7.2 Using the Web Admin User Interface

Use the Web Admin user interface to monitor and control your KiTSM system:

Even though all KiTSM processes are green and running the state of the KiTSM Server must be checked.

Perform the following steps:

- Start the KiTSM Administration GUI (<http://localhost:7420/TsmSAdmin/logon.do>).
- You need now to log in as:
 - **Tenant:** KiAdminDomain
 - **Role:** KiClusterAdmin
 - **Login:** KiClusterAdmin
 - **Password:** 123abc

16.8 Monitoring General KiTSM Server State

A KiTSM Server has two states: a **TSM STATE** and an **ADM STATE**.

The **TSM STATE** is responsible for the green run field in the WSM Service Monitor. It indicates that the server process is physically running.

To use the running KiTSM server with WISKI7 the **ADM STATE** must be set to **ON**.

Normally this is the case, however it is worth to verify this.

This check is also necessary, if WISKI7 returns the error message **NO FREE TSM** while logging in to KiTSM.

TSM CLUSTER > CLUSTER NODES

Check the **ADM STATE** entry of your KiTSM Server.

Home	TSM Cluster	Log Viewer	Tenants	Authorisation	Preferences					
Controller	Cluster nodes	Cluster load	Dispatcher	Cluster sessions						
<div><div></div><div></div><div></div><div>1 - 7 / 7</div><div></div><div></div><div></div></div>										
<div><div></div><div></div></div>	<div>Key</div> <div>99001</div>	<div>Instance id</div> <div>906001</div>	<div>Host</div> <div>192.168.54.129</div>	<div>Instance name</div> <div>1</div>	<div>Service name</div> <div>tsm-1</div>	<div>State</div> <div>on</div>	<div>Adm State</div> <div><div></div> on</div>	<div>Heartbeat Date</div> <div>21.01.2013 14:28:06 CET</div>	<div>Tolerance [min]</div> <div>2</div>	<div>Lo</div> <div>10</div>
Applications in cluster node										
	<div>Key</div>	<div>Type</div>	<div>Name</div>	<div>IP Port</div>	<div>State</div>	<div>Interactive Usage</div>	<div>Sessions</div>	<div>Tasks</div>		
<div><div></div><div></div></div>	<div>129001</div>	<div>kiiosys</div>	<div>KilOSystem</div>	<div>7443</div>	<div>on</div>	<div><div></div> on</div>	<div>0</div>	<div>0</div>		
<div><div></div><div></div></div>	<div>128001</div>	<div>tsmserver</div>	<div>tsmserver</div>	<div>7442</div>	<div>on</div>	<div><div></div> on</div>	<div>60</div>	<div>0</div>		
<div><div></div><div></div></div>	<div>98001</div>	<div>905001</div>	<div>qs-perf-2.kisters.de</div>	<div>1</div>	<div>tsm-1</div>	<div>on</div>	<div><div></div> on</div>	<div>21.01.2013 14:28:17 CET</div>	<div>2</div>	<div>8</div>
Applications in cluster node										
	<div>Key</div>	<div>Type</div>	<div>Name</div>	<div>IP Port</div>	<div>State</div>	<div>Interactive Usage</div>	<div>Sessions</div>	<div>Tasks</div>		
<div><div></div><div></div></div>	<div>127001</div>	<div>kiiosys</div>	<div>KilOSystem</div>	<div>7443</div>	<div>on</div>	<div><div></div> on</div>	<div>0</div>	<div>0</div>		
<div><div></div><div></div></div>	<div>126001</div>	<div>tsmserver</div>	<div>tsmserver</div>	<div>7442</div>	<div>on</div>	<div><div></div> on</div>	<div>60</div>	<div>0</div>		
<div><div></div><div></div></div>	<div>97001</div>	<div>903001</div>	<div>qs-perf-1.kisters.de</div>	<div>20</div>	<div>operation-1</div>	<div>on</div>	<div><div></div> on</div>	<div>21.01.2013 14:28:26 CET</div>	<div>2</div>	<div>0</div>

In the steps before the Perl script `initdataw7.pl` all created the tenant database schema. This database schema contains the KiTSM and Dispatcher tables as well as the WISKI7 database tables. The WISKI7 tables are generated through `DbMaintain.exe`.

This tenant schema must be **ACTIVE** for the Dispatcher. If the desired tenant is not **ACTIVE**, WISKI7 returns a **NO OPEN TENANT DATABASE** error while logging in to KiTSM.

TENANTS > TENANTS

Check if the tenant state is **ONLINE**.

Home	TSM Cluster	Log Viewer	Tenants	Authorisation	Preferences
Application Db plugins	Groups	Tenants			
1 - 1 / 1					
Id	Tenant	State	Session duration	Groups	Template
Id	Database storage	State	Extra db	JDBC URL	Db user
Id	Application-db	State	Exists	Version up to date	Version
Id	Application-db	State	Exists	Version up to date	Version
± 1001	MaiDomain	online	300		
1001	tsmDb	online	no	jdbc:sqlserver://qs-db-srv-1:1433;DatabaseName=tiwag_kitsm3	wiskiadm
1001	tsmserver	available	yes	yes	10_03
2001	kiiosys	available	yes	yes	2_06
2001	tcaExDb	online	yes	jdbc:sqlserver://qs-db-srv-1:1433;DatabaseName=tiwag_kitsm3	wiskiadm

16.9 Monitoring KiTSM Calculation Queues

KiTSM offers three types of background processes:

- Calculation trigger marking
- Execution of pending calculation
- Scheduled calculation processing

Note: These background processes were named Calculation Server in the WISKI6 system.

The **calculation trigger marking** process receives the changes in the time series data and translates these changes into update entries for all dependent time series. The **Execution of pending calculation** process then actually updates the time series data reflecting the change in source data. The **Scheduled calculation processing** background process is responsible for executing the scheduled calculations at a certain time.

The KiTSM System administration provides three queues, organised in tabs, for these background processes:

- Calculation trigger
- Pending calculations
- Scheduled calculations

Use the tab sequence **TENANTS, TENANTS** (select your tenant) > **TENANT: <YOUR TENANT>** > **CALCULATION TRIGGER** to monitor all calculations still in progress.

When you e.g. create or modify an agent, the trigger checks all depending time series and flags those for recalculation which are necessary.

Home

TSM Cluster

Log Viewer

TSM Domains

Authorisation

Preferences

Extra database

TSM Domains

TSM Domain: MaiDomain

TSM Domain

Background processes

Calculation trigger

Pending calculations

Scheduled calculations

Alarm buffer

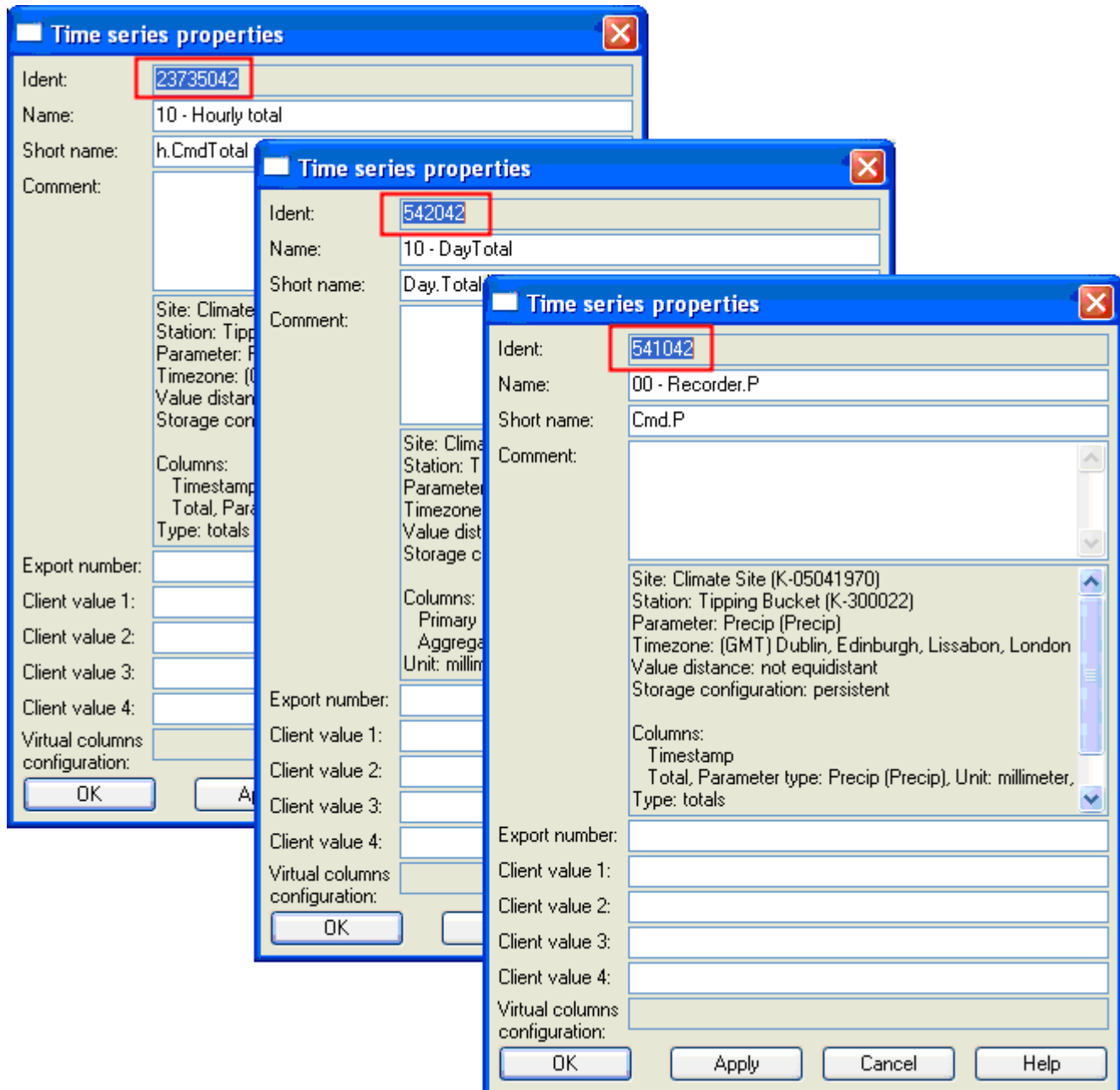
Alarm manager configuration

Domain

1 - 3 / 3

Key	Process	State	Caused by	Mark self	Exec origin	Error count	TsId	From	Until
			Error					Created	Started
36169042	new external event		yes	yes	0	541042	01.11.1982 00:00:00 CET	26.07.2006 00:59:59 CEST	
							29.03.2012 16:16:14 CEST		
36170042	new external event		yes	yes	0	542042	01.11.1982 00:00:00 CET	26.07.2006 00:59:59 CEST	
							29.03.2012 16:16:14 CEST		
36171042	new external event		yes	yes	0	23735042	01.11.1982 00:00:00 CET	26.07.2006 00:59:59 CEST	
							29.03.2012 16:16:14 CEST		

Note: The TsId is the internal time series identifier. You can verify it in the WISKI7 time series properties (to be opened with the Properties context menu item on the desired time series in the Explorer list view):



Use the tab sequence **TENANTS**, **TENANTS** (select your tenant) > **TENANT: <YOUR TENANT>** > **PENDING CALCULATIONS** to monitor all calculations still in progress.

The screenshot displays the WISKI Web Admin interface. The top navigation bar includes tabs for 'Home', 'TSM Cluster', 'Log Viewer', 'TSM Domains', 'Authorisation', and 'Preferences'. The 'TSM Domains' tab is active, showing a sub-navigation bar with 'Extra database', 'TSM Domains', and 'TSM Domain: MaiDomain'. The main content area is titled 'Pending calculations' and shows a table of calculation entries. The selected entry has a key of 20009042, a state of 'error', and an error count of 298. The error message is detailed, showing a NullPointerException during the execution of an agent. The interface also includes a sidebar with icons for various actions like 'Key', 'Process', 'State', 'Error count', and 'Tsid'.

You can set the status of the calculation request back to new; click the  button for that. With this change the entry will be put back to automatic update. Before changing the state, the cause of the error should be purged.

Click the  button to set the state of an entry (from new) to error. With this change the entry will not be selected for future calculations of automatic update.

Note: Already running processes will not respect this state change.

Click the  button to change the state of multiple erroneous entries at a time. With this button a screen is selected where all entries in error state or all entries of one time series identifier in error state can be reset to the new state.

Click the  button to expand the time series path for the time series identifiers of the entries. The language independent path and the path in the selected language are displayed. Click the  button to hide the time series path for the time series identifiers of the entries.

Click the  button to limit the time range of the processed calculation entries to the current time stamp. Click the  button to unlimit the time range again.

Use the tab sequence **TENANTS, TENANTS** (select your tenant) > **TENANT: <YOUR TENANT>** > **SCHEDULED CALCULATIONS** to monitor all calculations that were organised with a schedule. **Note** that WISKI7 provides a special time series property for that. This feature is optional. **Note** that this feature can be used to reduce the workload of your system!

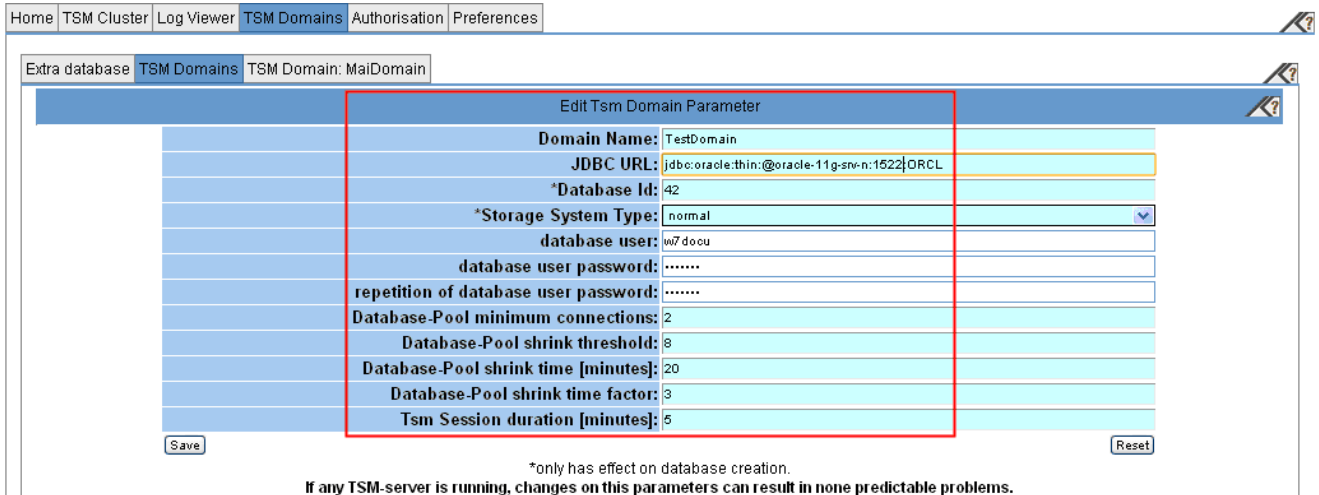
You can use the Web Admin user interface to perform e.g. these system management and system optimisation tasks:

- How to add a new tenant
- How to determine system load

16.9.1 Adding a New Tenant

You can add new tenants to your current setup.

- Use the tab sequence **TENANTS, TENANTS**.
- Click the  button to add a new KiTSM tenant configuration.
- Enter the relevant connection parameters and confirm with **[SAVE]**.



Home | TSM Cluster | Log Viewer | **TSM Domains** | Authorisation | Preferences

Extra database | TSM Domains | TSM Domain: MaiDomain

Edit Tsm Domain Parameter

Domain Name: TestDomain

JDBC URL: jdbc:oracle:thin:@oracle-11g-srv-n:1522:ORCL

*Database Id: 42

*Storage System Type: normal

database user: w7docu

database user password:

repetition of database user password:

Database-Pool minimum connections: 2

Database-Pool shrink threshold: 8

Database-Pool shrink time [minutes]: 20

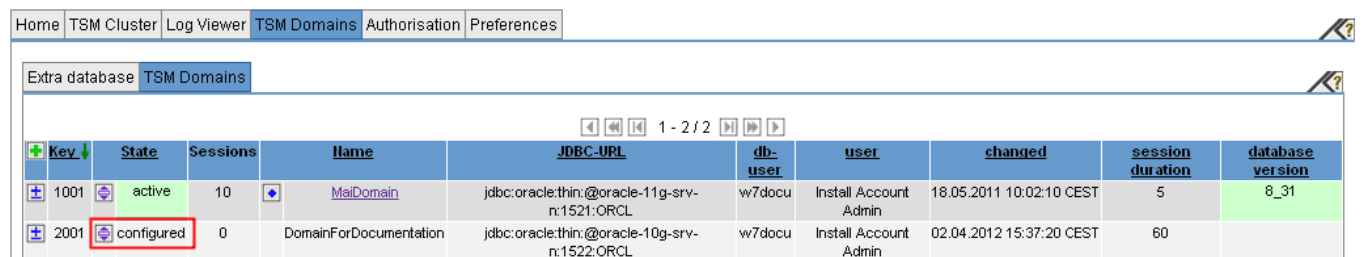
Database-Pool shrink time factor: 3

Tsm Session duration [minutes]: 5

Save Reset

*only has effect on database creation.
If any TSM-server is running, changes on this parameters can result in none predictable problems.

Your new tenant is added to the tenant list.



Home | TSM Cluster | Log Viewer | **TSM Domains** | Authorisation | Preferences

Extra database | TSM Domains

1 - 2 / 2

Key	State	Sessions	Name	JDBC-URL	db-user	user	changed	session duration	database version
1001	active	10	MaiDomain	jdbc:oracle:thin:@oracle-11g-srv-n:1521:ORCL	w7docu	Install Account Admin	18.05.2011 10:02:10 CEST	5	8_31
2001	configured	0	DomainForDocumentation	jdbc:oracle:thin:@oracle-10g-srv-n:1522:ORCL	w7docu	Install Account Admin	02.04.2012 15:37:20 CEST	60	

The state is set to configured.

- Click the  icon in the **STATE** column.
- Click the **[CREATE TENANT DATABASE]** button in the TsmDomainState tab to create the new tenant database.

Your new tenant will be created.

16.9.2 How to Determine and Control System Load

One tool to monitor your system, e.g. in case of heavy workload or high memory consumption (CPU usage), is to check the processes related to KiTSM and WISKI.

Use the Windows Task Manager to identify the processes by their PID (process ID):

The screenshot displays two windows side-by-side. The left window is the Windows Task Manager, showing a list of processes. The right window is the WSM Service Monitor, showing a list of services.

Task Manager Processes:

Name	PID	Username	CPU	CPU time	Memory (a...)	Command line
java.exe	26468	techdok	04	00:00:31	2,373,224 K	"C:\kisters\wisk\services\elasticsearch\bin\...\java\lt
java.exe	7420	techdok	10	104:30:38	653,368 K	C:\kisters\wisk\services\dcserver\...\java\bin\java -Xrs
java.exe	24440	techdok	03	00:07:37	555,952 K	"C:\kisters\wisk\services\java\bin\java" -Xms256M -X
java.exe	7340	techdok	02	27:16:06	396,904 K	"C:\kisters\wisk\services\java\bin\java.exe" -Djava.uti
java.exe	9360	techdok	00	01:20:52	287,172 K	C:\kisters\wisk\services\dcserver\...\java\bin\java -Xrs
java.exe	10196	techdok	00	01:07:46	292,412 K	C:\kisters\wisk\services\dcserver\...\java\bin\java -Xrs
java.exe	1888	techdok	01	01:48:18	293,176 K	C:\kisters\wisk\services\jprocontroller\...\java\bin\jav
java.exe	9784	techdok	00	01:05:30	285,464 K	C:\kisters\wisk\services\dcserver\...\java\bin\java -Xrs
java.exe	8200	techdok	00	00:45:38	169,160 K	C:\kisters\wisk\services\kidsm\...\java\bin\java -Xmx1;
java.exe	9068	techdok	00	00:23:40	107,140 K	C:\kisters\wisk\services\kidsm\...\java\bin\java -Xmx1;
java.exe	1256	techdok	00	00:41:18	273,832 K	"C:\kisters\wisk\services\java\bin\java" -Xms256M -X
java.exe	3008	techdok	00	01:06:33	471,264 K	C:\kisters\wisk\services\dcserver\...\java\bin\java -Xrs
java.exe	6744	techdok	00	00:26:49	168,968 K	C:\kisters\wisk\services\kidsm\...\java\bin\java -Xmx2;
java.exe	8116	techdok	00	00:28:44	258,780 K	C:\kisters\wisk\services\jprocontroller_kidsm\...\java\
java.exe	8884	techdok	00	00:22:19	116,724 K	C:\kisters\wisk\services\kidsm\...\java\bin\java -Xmx1;
java.exe	9228	techdok	01	01:22:28	525,140 K	C:\kisters\wisk\services\dcserver\...\java\bin\java -Xrs
java.exe	8612	techdok	00	00:36:36	110,520 K	C:\kisters\wisk\services\kidsm\...\java\bin\java -Xmx1;
java.exe	6012	techdok	00	00:34:37	176,176 K	C:\kisters\wisk\services\kidsm\...\java\bin\java -Xmx1;
java.exe	9292	techdok	01	00:06:14	247,324 K	C:\kisters\wisk\services\processanalytics\bin\...\java\
java.exe	10896	techdok	00	00:02:13	154,156 K	C:\kisters\wisk\services\kidsm\...\java\bin\java -Xmx5
java.exe	12876	techdok	00	00:00:27	20,844 K	"java" -cp ./rmiService/./ReportEngine/lib/* de.kisters
javaw.exe	5616	techdok	07	13:58:45	458,448 K	"C:\kisters\wisk\servermanager\java\bin\javaw.exe" ->

WSM Service Monitor:

Name	PID	Status
WISKI Server [localhost:1099]		
WebServer-Node-0	6560	Running
Elasticsearch	26468	Running
wiski-data-provider	20836	Running
timeseries-validation	1256	Running
tsmdispatcher	7420	Running
dispatcher-db		Running
queueserver	9360	Running
automation	9784	Running
interactive	9228	Running
scriptserver	3008	Running
tsm1	10196	Running
Services-Node-0	24140	Running
ProcessAnalyticsImporter	9292	Running
KIDSM Server [localhost:8999]		
KIDSM ConnectorServer	6744	Running

Check the CPU usage as known from your Windows system.

Note the **user name** of the processes related to KiTSM and WISKI: They differ from the default system user name (usually **Administrator** or **KISTERS**). This makes it possible to identify them also by the user name.

16.10 Solve performance issues on Windows 2022 with RDP connections

Due to a bug in the Windows 2022 Tablet service WISKI has some significant performance issues with RDP connections. This can be fixed by deactivating the Tablet service:

- Delete the sub keys of **HKEY_LOCAL_MACHINE\SYSTEM\Software\Microsoft\TIP\TestResults**
- Disabled the **TabletInputService** service (Touch Keyboard and Handwriting Panel Service) in the services app
- In the services tab of the task manager identify the PID of the **TabletInputService** and kill the service
- Take ownership of **C:\Program Files (x86)\Common Files\Microsoft Shared\ink\TabTip32.exe** and rename it (to prevent it from starting up and adding to **TestResults** key)
- Take ownership of **C:\Program Files\Common Files\microsoft shared\ink\TabTip.exe** and rename it (to prevent it from starting up and adding to **TestResults** key)
- Reboot the server

See also: [Server 2022 and Teams Bloats HKEY_LOCAL_MACHINE\SYSTEM\Software\Microsoft\TIP\TestResults\](#) - Microsoft Q&A

17 Optimising Your System

You can optimise WISKI7 and the WISKI server in manifold ways.

The next sub chapters will explain some optimisation steps within the Web Admin user interface:

- [Managing Background Processes](#)¹⁷⁵
- [Optimising Your System For Memory and CPU Usage](#)¹⁷⁶
- [Optimising Background Processes/Queues](#)¹⁷⁷

17.1 Managing Background Processes

The number of queues is unlimited. But the more queues you have, the slower your system could get. Since your system comes with some hardware restrictions, it is important to deal with the questions: How many CPU cores does your Server has and how large your working memory (RAM) is.

You can thus optimise your system regarding workload by managing your background processes.

Use the tab sequence:

TENANTS > TENANTS (select your required tenant) > **KITSM TENANT:** (your tenant name) > **BACKGROUND PROCESSES**

The screenshot shows the KISTERS Web Admin interface. The top navigation bar includes 'Home', 'TSM Cluster', 'Log Viewer', 'Tenants', 'Authorisation', and 'Preferences'. The 'Tenants' tab is selected, showing 'KITSM tenant: MaiDomain' and 'Tenant administration: MaiDomain'. Below this, the 'Background processes' tab is active. It displays a table with columns: Key, Type, active, TSM-Server, Server-Id, Number, Threads, and Min. memory. Two processes are listed: 118042 (calculation trigger marking) and 119042 (Execution of pending calculation). Each process has an 'Instances' sub-table with columns: Key, State, Session, update time, created, and Usage. A red box highlights the '+' button next to the process key 118042.

Key	Type	active	TSM-Server	Server-Id	Number	Threads	Min. memory
118042	calculation trigger marking	yes	192.168.55.73#1	1001	1		150
Instances							
Key	State	Session	update time	created	Usage		
1290042	active	1205001	29.01.2013 15:37:15 CET	29.01.2013 12:23:17 CET	7.690407E-4		
119042	Execution of pending calculation	yes	192.168.55.73#1	1001	1		150
Instances							
Key	State	Session	update time	created	Usage		
1289042	active	1204001	29.01.2013 15:37:15 CET	29.01.2013 12:23:17 CET	0.0021796403		

KISTERS logo is visible at the bottom left. The bottom right shows 'install user Cluster Admin', 'KiAdminDomain', and '29.01.2013 15:37:47 CET'.

Obey roughly to this rule of thumb: Install two working instances per CPU core. This means that on e.g. a quad core system you can have up to eight calculation queues for each background process – which means in total 24, if you have a system setup with these background processes:

- Calculation trigger
- Pending calculations
- Scheduled calculations

Click the button of the respective background process to modify the number of instances of the process.

Home TSM Cluster Log Viewer **Tenants** Authorisation Preferences

Application Db plugins Groups Tenants **KiTSM tenant: MaiDomain** Tenant administration: MaiDomain

KiTSM tenant **Background processes** Background calculation History migration Tenant properties

Change Background Processes

Tenant: MaiDomain
 Key: 118042
 Type: calculation trigger marking:
 Active: yes
 Number: 1
 Minimum free memory [MB]: 150

Save Reset

KISTERS install user Cluster Admin KiAdminDomain 29.01.2013 22:30:59 CET

The result could look like this:

Home TSM Cluster Log Viewer **TSM Domains** Authorisation Preferences

Extra database TSM Domains **TSM Domain: MaiDomain**

TSM Domain **Background processes** Calculation trigger Pending calculations Scheduled calculations Alarm buffer Alarm manager configuration Domain properties

Key	Type	active	TSM-Server	Server-Id	Number	Min. memory
115042	calculation trigger marking	yes	192.168.55.73#1	81001	1	150
Instances						
Key	State	Session	update time	created	Usage	
1128042	active	8159001	30.03.2012 11:51:44 CEST	29.03.2012 14:25:31 CEST	9.771538E-5	✕ ✕
116042	Execution of pending calculation	yes	192.168.55.73#1	81001	8	150
Instances						
Key	State	Session	update time	created	Usage	
1129042	active	8167001		30.03.2012 11:52:12 CEST	0.0	✕ ✕
1130042	active	8168001		30.03.2012 11:52:12 CEST	0.0	✕ ✕
1126042	active	7961001	30.03.2012 11:51:44 CEST	22.03.2012 12:14:15 CET	2.3444237E-4	✕ ✕
1131042	active	8169001		30.03.2012 11:52:12 CEST	0.0	✕ ✕
1132042	active	8170001		30.03.2012 11:52:13 CEST	0.0	✕ ✕
1133042	active	8171001		30.03.2012 11:52:14 CEST	0.0	✕ ✕
1134042	active	8172001		30.03.2012 11:52:14 CEST	0.0	✕ ✕
1135042	active	8173001		30.03.2012 11:52:15 CEST	0.0	✕ ✕
117042	scheduled calculation processing	yes	192.168.55.73#1	81001	1	150
Instances						
Key	State	Session	update time	created	Usage	
1127042	active	7962001	30.03.2012 11:51:44 CEST	22.03.2012 12:14:15 CET	1.5763763E-4	✕ ✕

Note:

- The **USAGE** column indicates the workload of the respective processes. If the usage is zero or close to, there is nothing to do for the background processes (no imports running, no time series calculation necessary at the moment, etc.).
- This assumes that no other services need workload. So if e.g. a KiDSM is running on the same machine, reduce the number of calculation triggers and pending calculation processes to e.g. six to have two more for the KiDSM.
- Of course this is only a random access. You need to investigate whether your performance gets better or weaker with more or less calculation queues.
- Also take your import procedures into account: The more import processes you run in your system, the higher the workload!

17.2 Optimising Your System For Memory and CPU Usage

Since KiTSM uses a Java Virtual Machine, the processes of this machine must be equipped with the highest possible memory. Default is 768 MB but for larger KiTSM systems this value should be increased.

To set the KiTSM memory to a value which fits the system requirements of WISKI7, use as rule of thumb approximately 500 MB + 250 MB per worker/instance of a background process. So if four workers are doing their job, increase to approx. 1,500

MB of memory – which is the maximum possible value on 32-bit operating systems. This is why we recommend 64-bit operating systems.

Note: Avoid swap file usage! Do not forget to leave enough physical memory for your operating system!

Use this setting:

- Move to `\wiski\services\dcserver\server_tsm1\configuration`
- Open (or create if it does not exist) `dcserver-custom.properties`.
- Modify this entry:

```
vm.memory=1024
```

- Do not forget to uncomment this entry if it is currently commented out.
- Save the file and restart the KiTSM Controller service (either manually from the command line or through the Windows system service) to apply this setting to your KiTSM system.

17.3 Optimising Background Processes/Queues

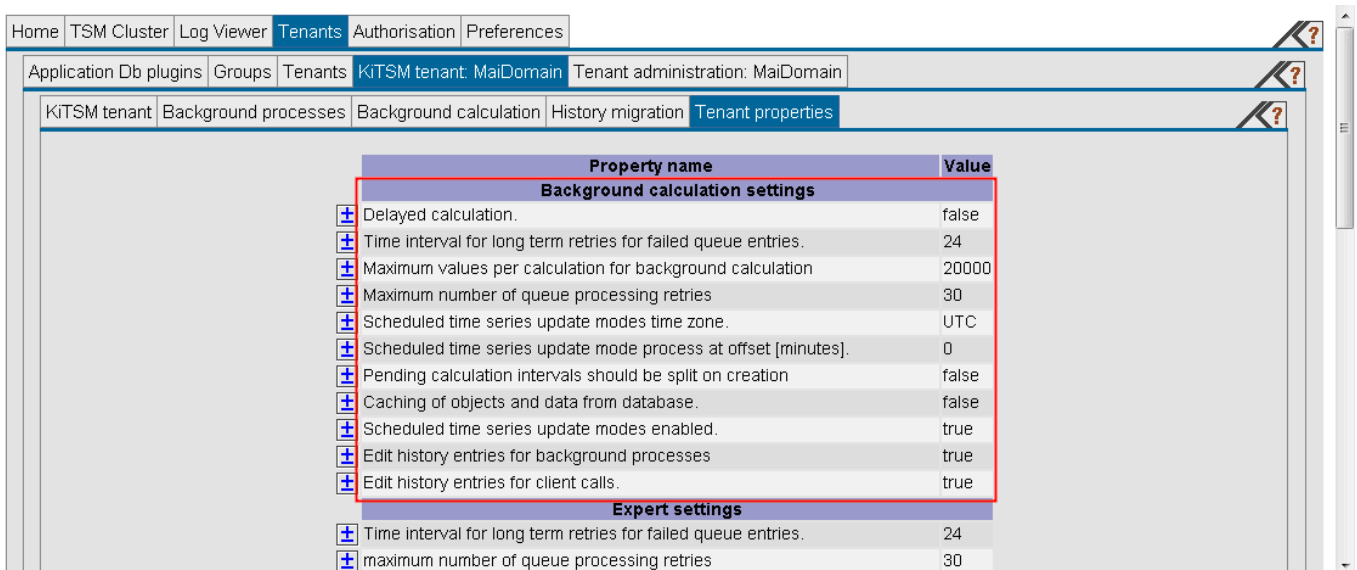
You can also optimise your system by reducing the workload within the pending calculation queues.

You can change multiple settings for the background processes and specify some elements for error handling in your calculation queues.

Use the tab sequence:

TENANTS > TENANTS (select your required tenant) > **KITSM TENANT:** (your tenant name) > **BACKGROUND PROCESSES.**

Click the  button to address the settings.



Property name	Value
Background calculation settings	
☐ Delayed calculation.	false
☐ Time interval for long term retries for failed queue entries.	24
☐ Maximum values per calculation for background calculation	20000
☐ Maximum number of queue processing retries	30
☐ Scheduled time series update modes time zone.	UTC
☐ Scheduled time series update mode process at offset [minutes].	0
☐ Pending calculation intervals should be split on creation	false
☐ Caching of objects and data from database.	false
☐ Scheduled time series update modes enabled.	true
☐ Edit history entries for background processes	true
☐ Edit history entries for client calls.	true
Expert settings	
☐ Time interval for long term retries for failed queue entries.	24
☐ maximum number of queue processing retries	30

Set the Pending calculations batch size under the **BACKGROUND CALCULATION SETTINGS** to a different value.

- This setting determines the number of queue entries that are handled by a background process at a time. Increase the default setting of 1 (which normally is too low!) if your workers are waiting most of the time!
- Start with value 5 to 50. Valid values are from 1 to 100.
- The optimum value depends on the system load, the number of background processes and the import block size and import frequency.

The screenshot shows the KISTERS web interface with the 'Tenants' tab selected. The 'Edit tenant property' dialog is open for the 'KITSM tenant: MaiDomain'. The 'Queue server settings' section is active, showing the 'Pending calculations batch size' property. The description states: 'Set the batch size for pending calculations - increase if the worker are waiting most of the time.' The value range is 'integer: 1..100'. The current value is '3'. A 'Save' button is present. Below the settings, a general information message reads: 'General Information: Please check the value before saving. The software only performs basic checks.'

- Under the **BACKGROUND CALCULATION SETTINGS** you can also manipulate these settings:
 - The maximum number of values per calculation (to restrict resource usage).
 - The Java time zone ID for scheduled time series update modes. A list of possible values can be found in the user preference screen.
 - The offset [min] for the process at date for scheduled time series update modes (0 = next day at 0:00 with respect to the configured time zone).
 - Whether to split pending calculation intervals on creation.
 - Whether to enable scheduled time series update modes.
 - Writing of edit history entries for background processes.
 - Writing of edit history entries for client calls.

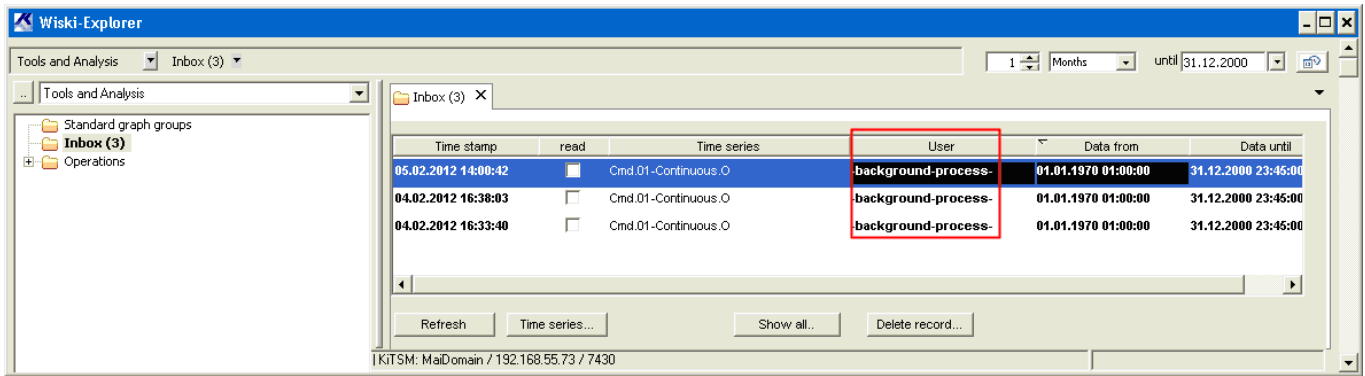
All actions on time series are saved in a edit history. This history can be influenced by these two settings:

- Edit history entries for background processes
- Edit history entries for client calls

You can change the history logging properties and **switch both settings off** so no more history is maintained (set the values to false).

The screenshot shows the KISTERS web interface with the 'Tenants' tab selected. The 'Edit tenant property' dialog is open for the 'KITSM tenant: MaiDomain'. The 'Background calculation settings' section is active, showing the 'Edit history entries for background processes' property. The description states: 'Determines writing of edit history entries for background processes.' The value range is 'false, true'. The current value is 'false'. A dropdown menu is open, showing the options 'false' and 'true'. Below the settings, a general information message reads: 'General Information: Please check the value before saving. The software only performs basic checks.'

Note: You can find the edit history entries also in your WISKI7 client application: all subscribed time series are logged in the Inbox of your WISKI7 Explorer.



So if you switch both settings off so no more history is maintained (by setting the values to false), the Inbox will remain empty.

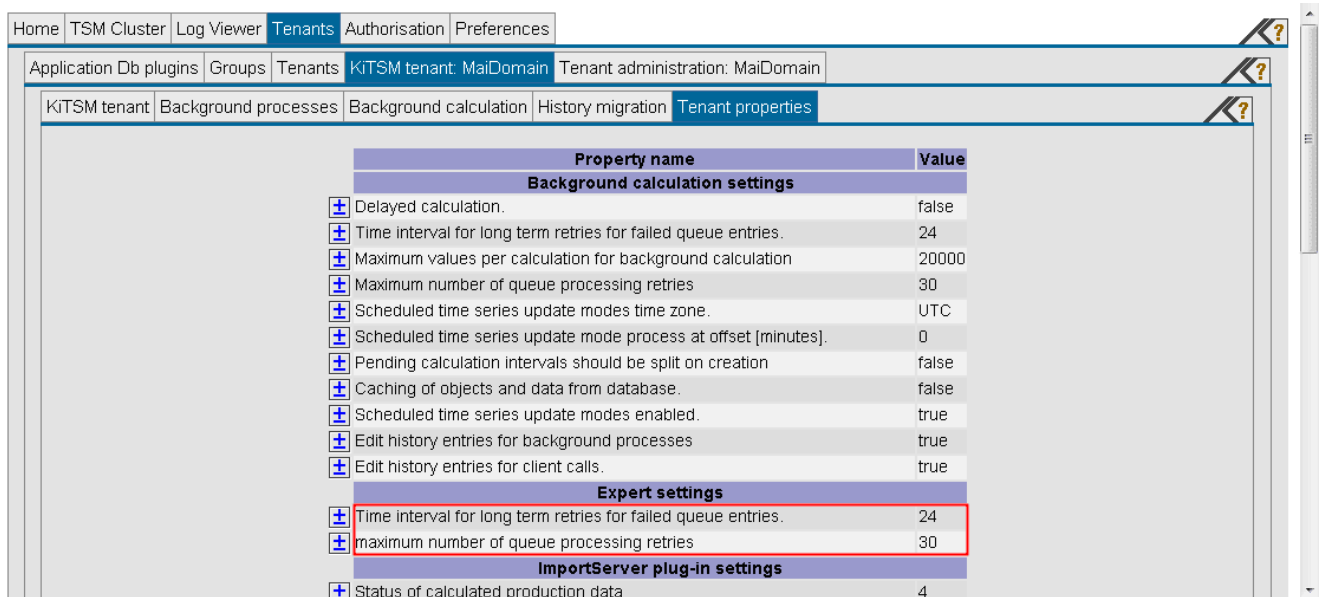
Or you can **delete** old history entries:

```
select * from ts_edithistory
delete from TS_EDITHISTORY where TSEH_DATE_UTC <
to_date('2011-11-01','YYYY-MM-DD')
```

Note: This can at the moment only be achieved by SQL select statements. So contact us if you are interested in more information on this issue.

Under the **EXPERT SETTINGS** section you can also manipulate these two settings to influence how often the system tries to recalculate erroneous entries in the calculation queues:

- the time interval in hours to retry processing of failed entries, even if the maximum number of retries is already reached.
- the maximum number of retries to process previously failed queue entries.



Note: These expert settings control the automatic recalculation of erroneous calculation requests. You can also set back failed queue entries manually:

Use the tab sequence:

TENANTS > TENANTS (select your tenant) > **TENANT:** (your tenant) > **PENDING CALCULATIONS**.

Click the  button of an entry to reset single erroneous entries.

Click the  button to change state of all erroneous entries.

Click the **[RESET STATE OF ERRONEOUS ENTRIES]** button to set back all existing faulty entries.

Two more ways of saving workload are in your WISKI7 client system:

- Think about to configure your monthly and annual time series as virtual time series that are not calculated all the time source time series change. So use virtual time series for those you only use occasionally.

- Daily time series which are not editable could be configured as scheduled time series in order to calculate them only once a day and hence reduce the workload. **Note:** This can at the moment only be achieved by SQL select statements. So contact us if you are interested in more information on this issue.

18 Using the WISKI Standard Portal

The WISKI7 Standard Portal (or simply 'Portal') has been available since WISKI7 version 7.4.7 SR14. For the best user experience, we recommend using the latest versions of Chrome, Firefox, Safari or Edge. Our software is tested and supported on several additional versions no older than 12 months. However, older versions may still work with limitations in animation performance and web layout/styling. Internet Explorer 11 is supported in the latest versions released in 2017 (11.0.44). In line with Microsoft, we no longer support earlier versions of Internet Explorer. Desktop computers should have sufficient RAM (i.e. 1 GB) and network bandwidth (i.e. ≥ 5 Mbps). The application takes advantage of JavaScript and does not require any additional browser extensions (plugins or add-ons).

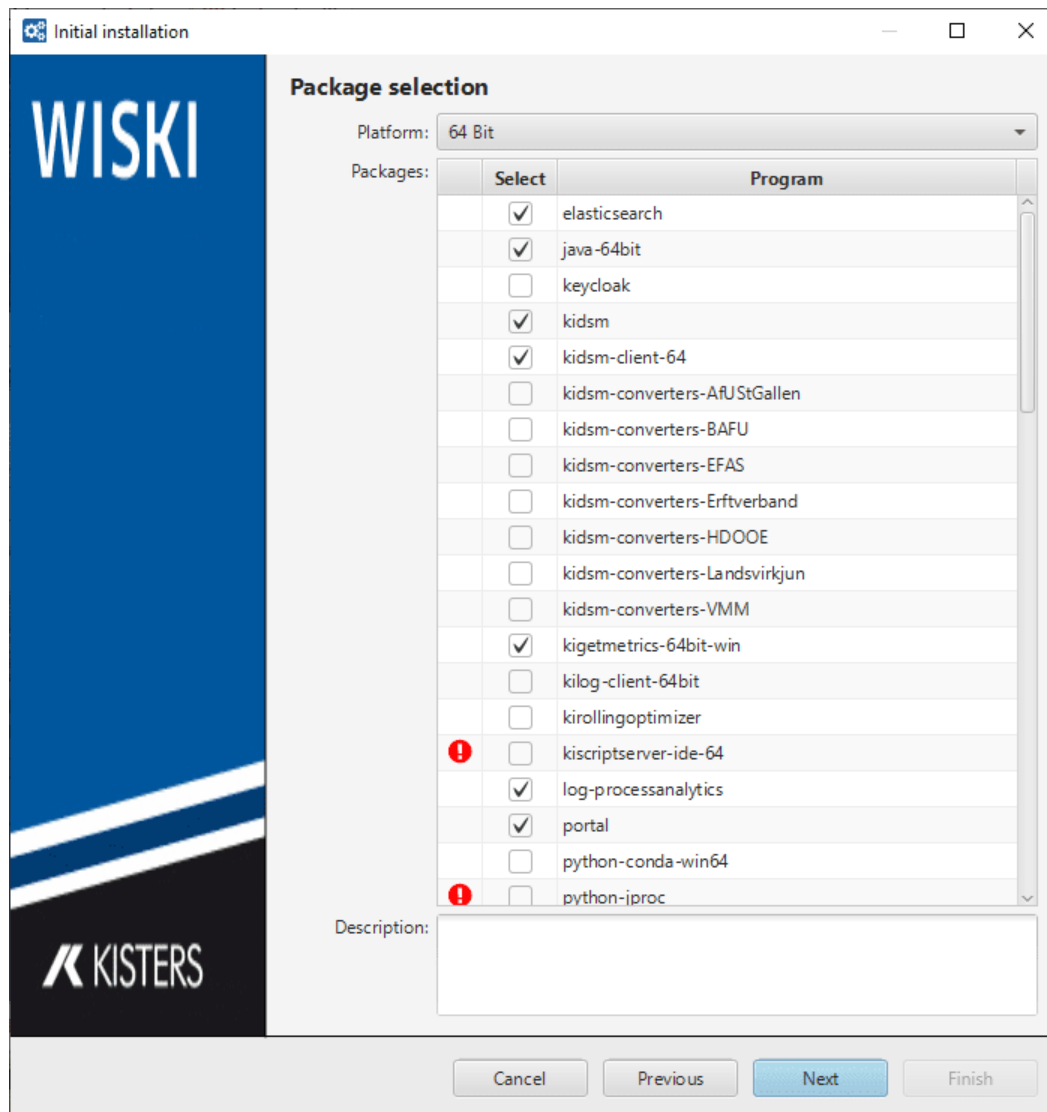
With the release of 7.4.11 SR7, both the WISKI7 and WISKI7 Standard Portal support HTTPS. For details on how to use the secure protocol, please refer to the section [Certificate and TLS Management in WISKI](#)^[123].

Detailed information can be found in the following subchapters:

- [Portal Installation](#)^[181]
- [Starting and Stopping](#)^[184]
- [Portal Administration \(since WISKI 7.4.11 SR7\)](#)^[185]
- [Troubleshooting for the Portal](#)^[186]
- [Standard Applications](#)^[195]

18.1 Portal Installation

To use the WISKI Standard Portal you need a certain set of packages. See [Package selection](#)^[65] for details.



Update installation - Install WINDOWS services

WISKI

☐ **WISKI Server** ☒ System ☐ Other
 User: Password:

☐ **KiDSM** ☒ System ☐ Other
 User: Password:

☐ **KiRollingOptimizer** ☒ System ☐ Other
 User: Password:

☒ **WISKIDataProvider** ☒ System ☐ Other
 User: Password:

KISTERS

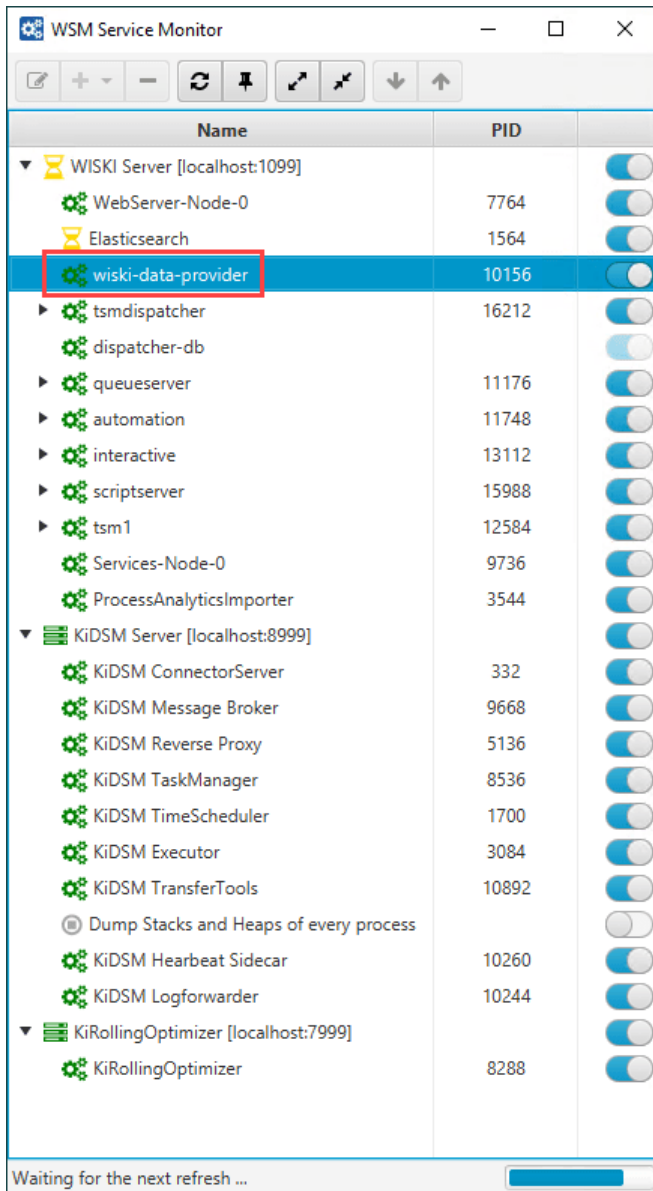
Cancel Previous **Next** Finish

After successful installation, the Admin user can add additional users via the WISKI Client (Guests role rights suffice). Existing WISKI users can log in at the Portal as long as their permissions allow access. When using Keycloak with LDAP authentication backend, users will be managed by the LDAP configuration selected. The WISKI Admin user can modify all permissions from within the Portal administration later, which will only affect rights set for the Portal.

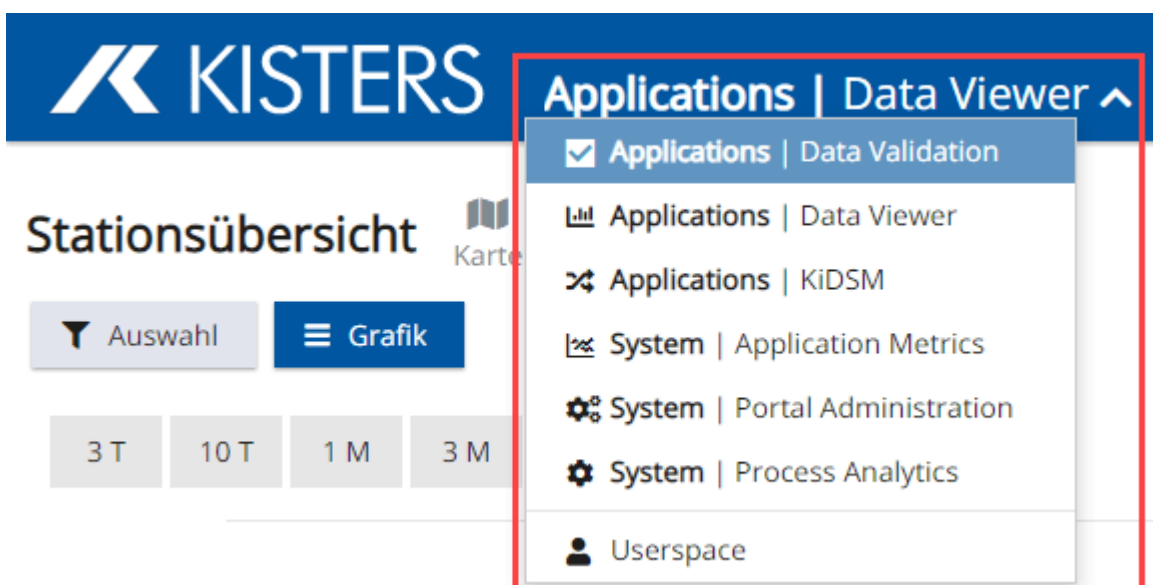
Note: After 7.4.11 SR7, the Data Viewer application will also be installed automatically with the selection of the **portal** package. Since that version, installations containing the Web applications Application Metrics and Data Viewer will also need the KiWIS and WDP packages (select the **kiwis** and the **wdp-win** package in the WISKI Server Manager). It is recommended to also install the WISKI Data Provider as a WINDOWS service* when WDP is selected – the WISKI Server Manager will ask for it.

*After 7.4.11 SR10, the WINDOWS service is not needed any longer. The WISKI Data Provider (WDP) will be included in the JProcController. Please select the **jproc** package for all portal installations to make sure the WDP is installed properly.

In the Service Monitor, you will see a node called **wiski-data-provider** after successful installation.

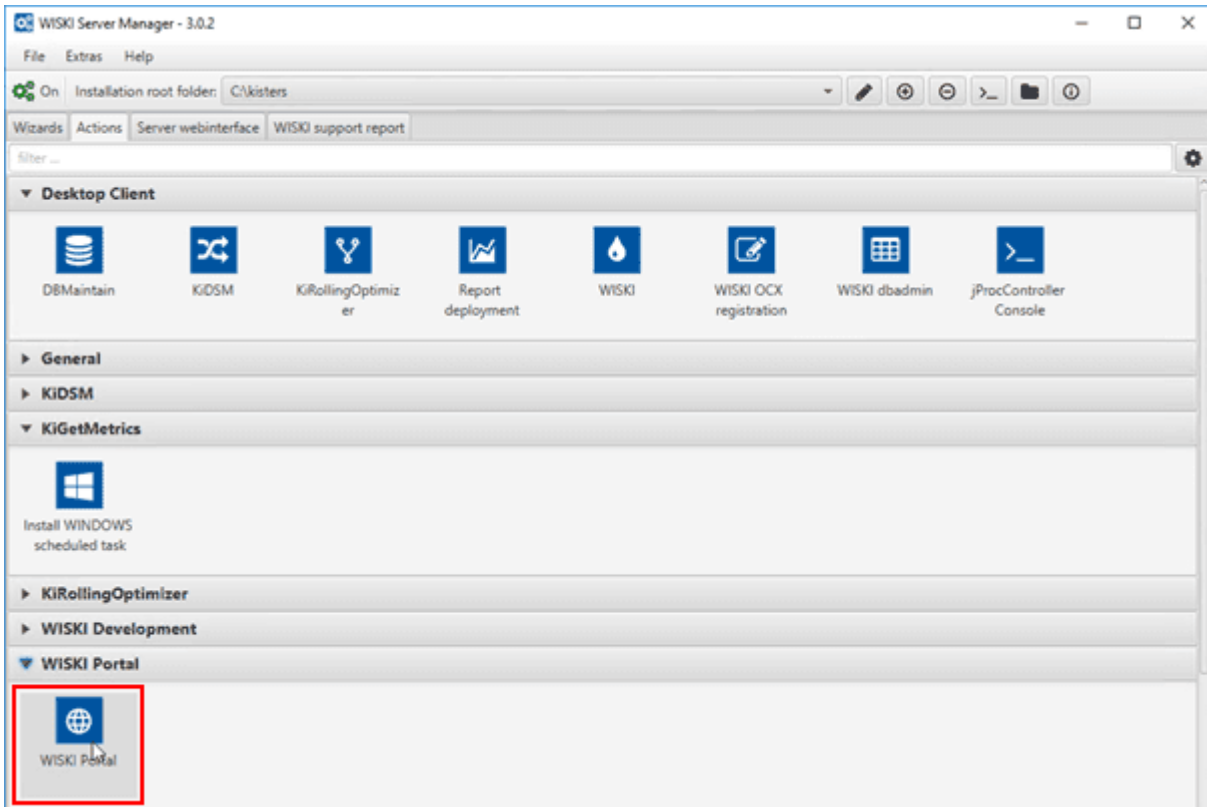


The WISKI Data Provider (WDP) serves as an API for data retrieval from WISKI by KiWIS and for the web applications based on it, such as the Data Viewer.



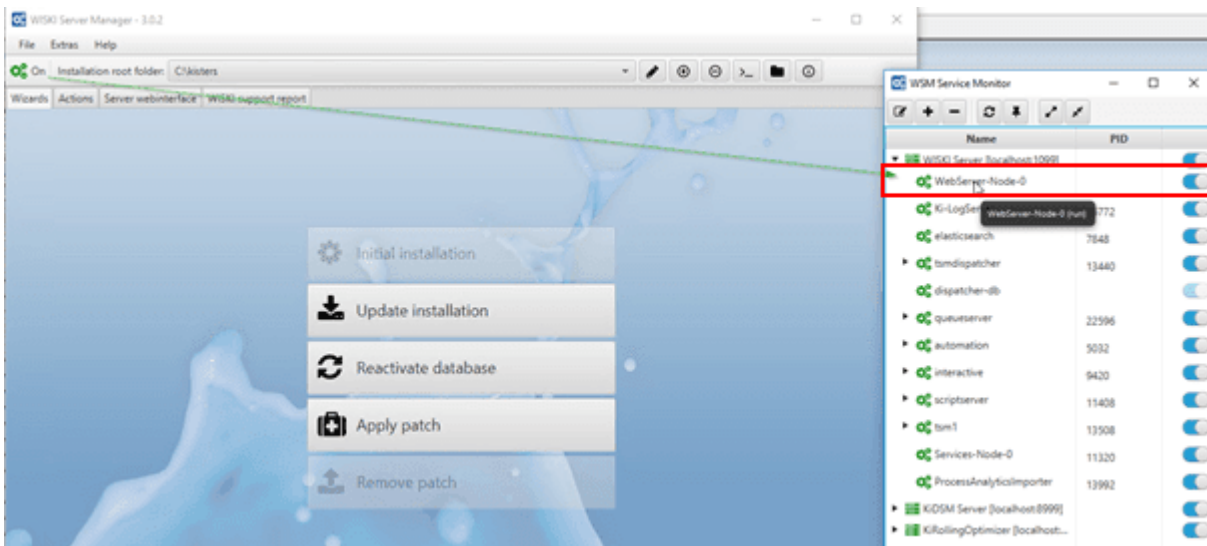
18.2 Starting and Stopping

To launch the Portal, double-click the **[WISKI PORTAL]** icon as shown below or visit <http://<hostname>:7481/applications> in your browser. For HTTPS, call <https://<hostname>:7482/applications> accordingly or use the **WISKI PORTAL** icon to start the action script, which will automatically detect the correct protocol and open the default browser.



To stop (or restart) the Portal web server, you can use the toggle button for **[WEBSERVER-NODE-0]**.

Note that the Portal applications need this node to be running (technically a Tomcat server).



18.3 Portal Administration (since WISKI 7.4.11 SR7)

The first login to the WISKI7 Standard Portal should be done with the standard WISKI7 **Admin** user. This user has the role **UserAdmin** and can see the **Portal Admin** application to manage users and portal settings.

General Settings

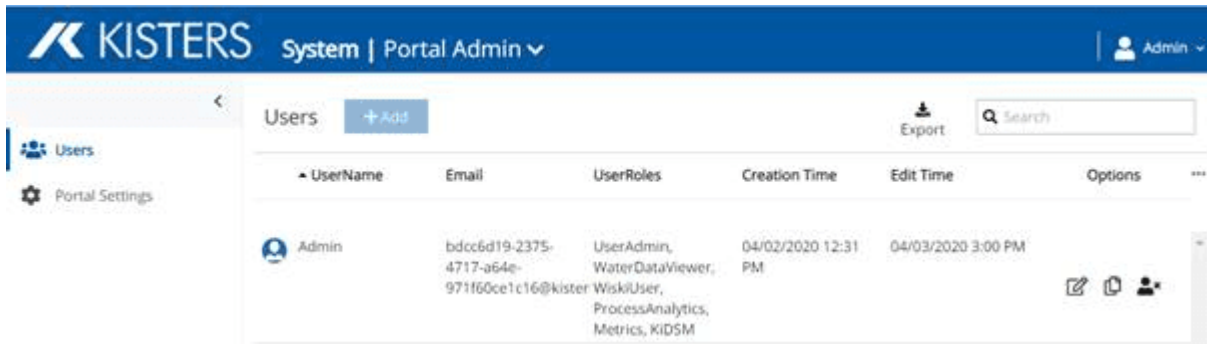
You can change the following base settings of the portal to your local preference:

Element	Description
Portal Title	This is used as a string visible in the header of tab pages or when the user stores the page as a bookmark / favourite.
Default Theme	KISTERS provides a larger set of different themes, mainly changing the color of the title bar as well as the navigation panel on the left hand side.
Language	The Standard Portal has several out-of-the box translations. When you select your preferred language, the portal also uses the date format which is typically used in this area.
Date Time Format	In case the language dependent data format does not fulfil your needs, you can individually select another date format here. Please check the language dependent format before using this individual option.
Time Zone	You may select a desired time zone for visualisation. If you leave this on DEFAULT all users will see the information in their individual time zone setting of the browser.
Default Application	Please select the default application that each user should see at first login. For all later logins the user will see the application he has used when he logged out the last time.
Application Selector Style	The application selector is the drop down list which appears when you click on the application title in the title bar (to the right of the KISTERS logo).
Logo Title	The logo title is the fly-over text appearing when you move with your mouse over the KISTERS logo at the top left.

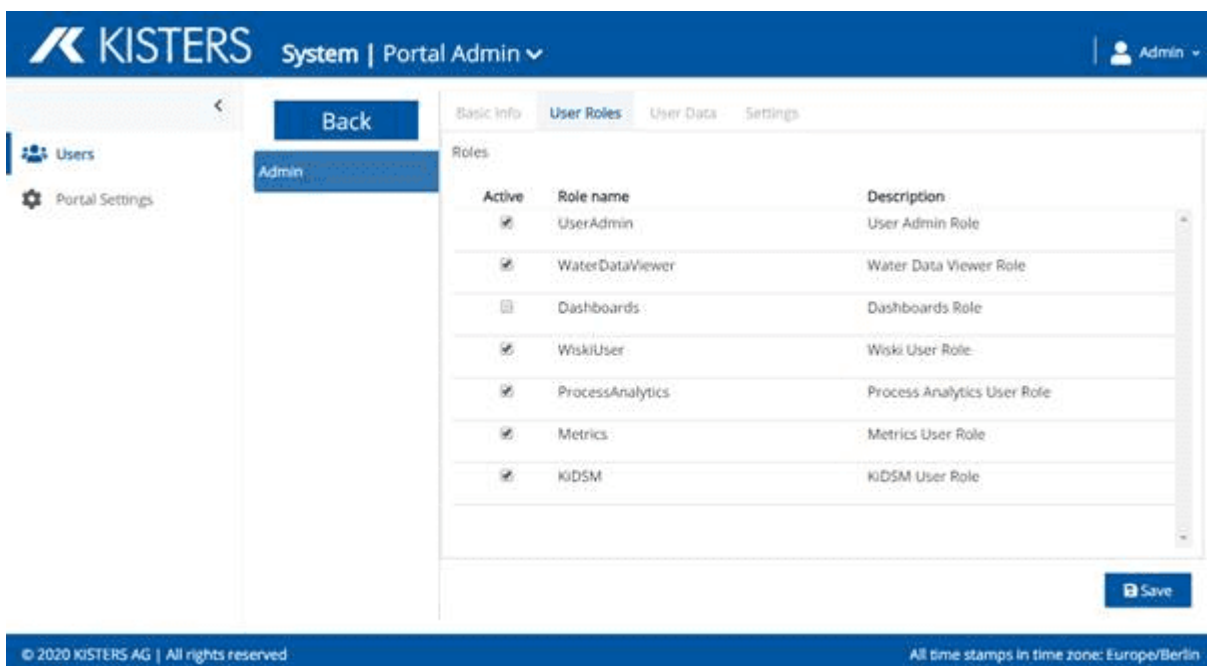
User Settings

The Portal Admin application provides the main user management system in the WISKI7 Standard Portal.

- When you want to create a new user, you need to create and configure them in WISKI7 first. Once a user has logged in at the portal (needed for registration), he/she will appear in the portal's user list like depicted below.



- Double-click on a user entry to manage the main user settings.
Note: The WISKI7 Standard Portal provides all major applications as roles. So you can easily maintain which user has access to which application.
- Click the **[SAVE]** button in the lower right corner to save your changes.



18.4 Troubleshooting for the Portal

The following checklist might help to get Keycloak running with the WISKI7 Standard Portal:

Versions with HTTP (before 04-2020, until 7.4.11 SR6)

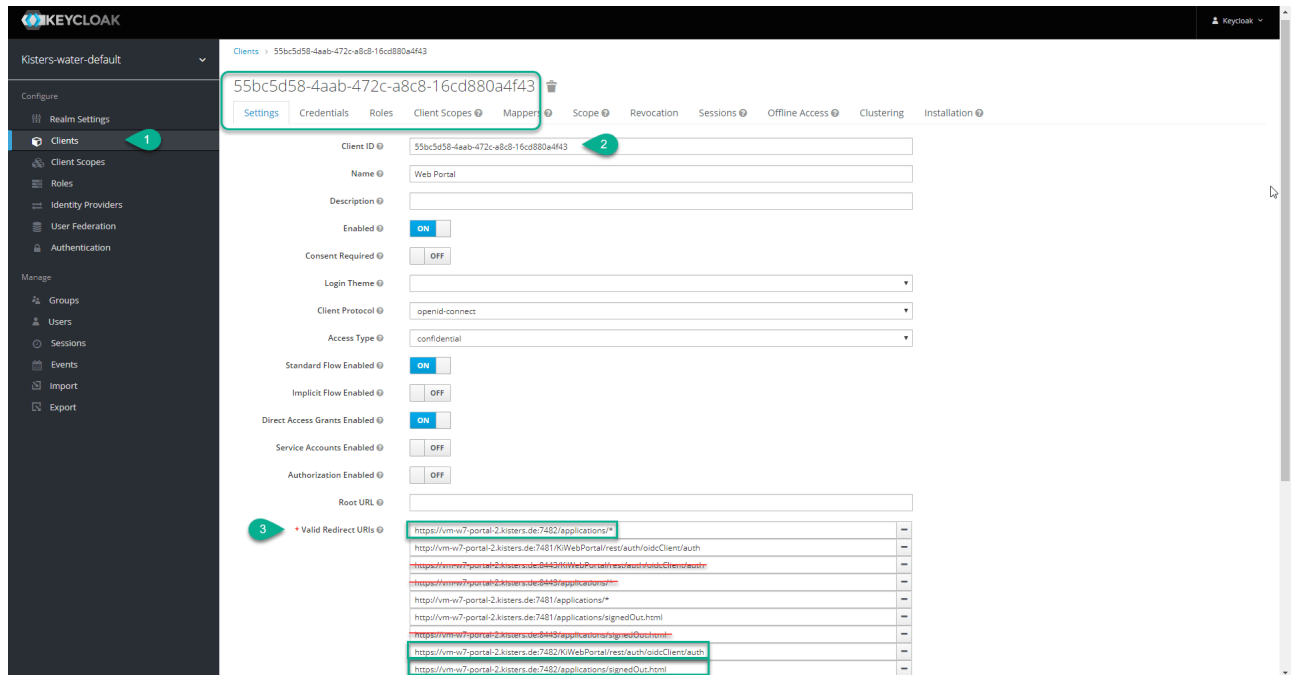
- Check if the PortalAdmin user exists in the portal-related database table (**WP_W7DIST_USER**).
- If not, run DB **maintenance.exe** and make sure the PortalAdmin gets the correct password for initialisation.
- Check if you can log in as that user in the WISKI client.
- Make sure the **webportal.properties** file matches the client settings (Client ID and encrypted Client Secret).
- Make sure that the LDAP settings are correct. Test with the buttons under **USER FEDERATION > ACTIVE DIRECTORY**, see screenshot below.

- Check the **MAPPERS** settings if you cannot log in with short name, try with mail / e-mail login, i.e. your LDAP credentials.
- If users are displayed in the WISKI client with their IDs instead of their user names, check if the mappings are set correctly in your Keycloak configuration (see description how to configure user federation mappers in [Configuring Keycloak for LDAP User Department Attribute](#)¹³⁷).
- Check these logs for errors:
 - `\wiski\services\tomcat\node0\logs\WebPortal.log`,
 - `\wiski\services\keycloak\data\log\keycloak.log`.
- If specific users do not have permissions on a fresh installation, check if the portal has been initialised properly. The PortalAdmin must be the **first** to login, then the other (LDAP) users will have permissions. Permission import settings can be checked in `\wiski\services\tomcat\node0\webapps\KiWebPortal\WEB-INF\conf\permImport\permImport.json`.

Versions with HTTPS (since April 2020, 7.4.11 SR7)

As a temporary workaround, for 7.4.11 SR7 and later, you need to perform a manual reconfiguration when using the portal and Keycloak after a WISKI7 update.

- Make sure that all nodes are running in the WSM Service Monitor, especially the WebServer-Node-0.
- Reconfigure Keycloak by fixing the redirect URIs (HTTPS on port 7482)
 - Visit `https://<canonical_hostname>:7415` and start the administration console. Log in with the username **keycloak** and the default password **12years**.
 - Navigate via Clients and Client ID **55bc5d58-4aab-472c-a8c8-16cd880a4f43** (for the Web Portal) to the **SETTINGS** tab.
 - Modify all valid redirect URIs with **https**: the port 8843 is obsolete, replace it by port 7482.



- Scroll down and save your settings.
- Run the portal installer script again with the following call from the folder `\wiski\services\portal\installer` (using `cmd`):
Note: You must have the correct PortalAdmin password. Replace the **XXX** in the following command with this password.

```
..\..\java\bin\java -jar portal-client-installer-standalone.jar --
password=XXX --keycloak=true --adminusername=Admin --
adminpassword=12years --excludeRoles=Dashboard
```

- You can add the flag `--verbose=true` for debugging purpose and with `--excludeRoles` you can add a comma-separated list of apps you do not want to be seen in the Portal.
- All users registered in the LDAP service can now log with their short name or e-mail address and their private password.

Versions with Data Validation (7.4.11 SR8 and higher)

Note: When using WISKI 7.4.11 **SR9** the first step can be skipped.

- Check if the TimeSeriesBuffer is turned OFF in the service registry, which indicates that
 - the registration was not done correctly and
 - tasks in DV do not work properly (you won't see any results after launching tasks).

Environment: Standalone | Data center: Dispatcher
Current time: 2020-06-17T11:09:19 +0200, Uptime: 16:19
Lease expiration enabled: true, Renew threshold: 40, Renew (last min): 164

Home | Last 1000 since startup
DS Replicas: [vm-w7-portal.kisters.de](#) |

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
DE.KISTERS.TIME.SERIES.BUFFER	n/a (1)	(1)	DOWN (1) - vm-w7-portal.kisters.de.7469/0
DE.KISTERS.TIME.SERIES.VALIDATION	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.8480/0
DE.KISTERS.TIME.SERIES.VALIDATION.REPOSITORY	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.8480/0
DE.KISTERS.TIME.SERIES.VALIDATION.RULES.IMPLEMENTATION	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7469/0
DE.KISTERS.WATER.AUTH.OPENIDCONNECT	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7414/7415
DE.KISTERS.WATER.DATABASE.ELASTICSEARCH	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.9200/0
DE.KISTERS.WATER.EXECUTIONQUEUE	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.8480/0
DE.KISTERS.WATER.KIWI	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.0/7482
DE.KISTERS.WATER.OPERATIONS.RESTFUL	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7469/0
DE.KISTERS.WATER.PORTAL	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7481/7482
DE.KISTERS.WATER.PUBLISHER	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7469/0
DE.KISTERS.WATER.REPORT.IMPLEMENTATION	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7469/0
DE.KISTERS.WATER.TIME.SERIES.ALGORITHMS.SOURCE	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7469/0
DE.KISTERS.WATER.WISKIHTTP	n/a (7)	(7)	UP (7) - vm-w7-portal.kisters.de.7406/7408, vm-w7-portal.kisters.de.7480/7479, vm-w7-portal.kisters.de.7401/7410, vm-w7-portal.kisters.de.7402/7415, vm-w7-portal.kisters.de.7403/7422, vm-w7-portal.kisters.de.7404/7423, vm-w7-portal.kisters.de.7405/7419
DE.KISTERS.WATER.WISKIHTTP.STATIC	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7480/7479
DE.KISTERS.WATER.WISKI.METADATA.EXPORT	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7480/7479
DE.KISTERS.WATER.WISKI.METADATA.ODATA	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7480/7479
DE.KISTERS.WATER.WISKI.RESTAPI	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7406/7408
DE.KISTERS.WATER.WORKFLOWENGINE	n/a (1)	(1)	UP (1) - vm-w7-portal.kisters.de.7470, de.kisters.water.workflowengine.7470

General Info

Name	Value
environment	standalone
num-of-cpus	4
vm-w7-portal.kisters.de.7469	7469mb

- For recent updates, you can quickly check the OIDC settings against the file `wiski\services\water-python-components\bin\.configuration\.openidconnect-TimeSeries Buffer`:
- To log in at the Admin Console (OIDC settings), see 2.a below
- Create a new client for the TimeSeriesBuffer.
 - Log in at the Keycloak admin console `<hostname>:7415` with user name `keycloak` and password `12years`.
 - Add your hostname temporarily to **Trusted Hosts**.
 - Use the interface **REALM SETTINGS > KISTERS WATER DEFAULT > CLIENT REGISTRATION > CLIENT REGISTRATION POLICIES > TRUSTED HOSTS > EDIT**

KEYCLOAK

Kisters-water-default

General Login Keys Email Themes Cache Tokens Client Registration Security Defenses

Initial Access Tokens Client Registration Policies

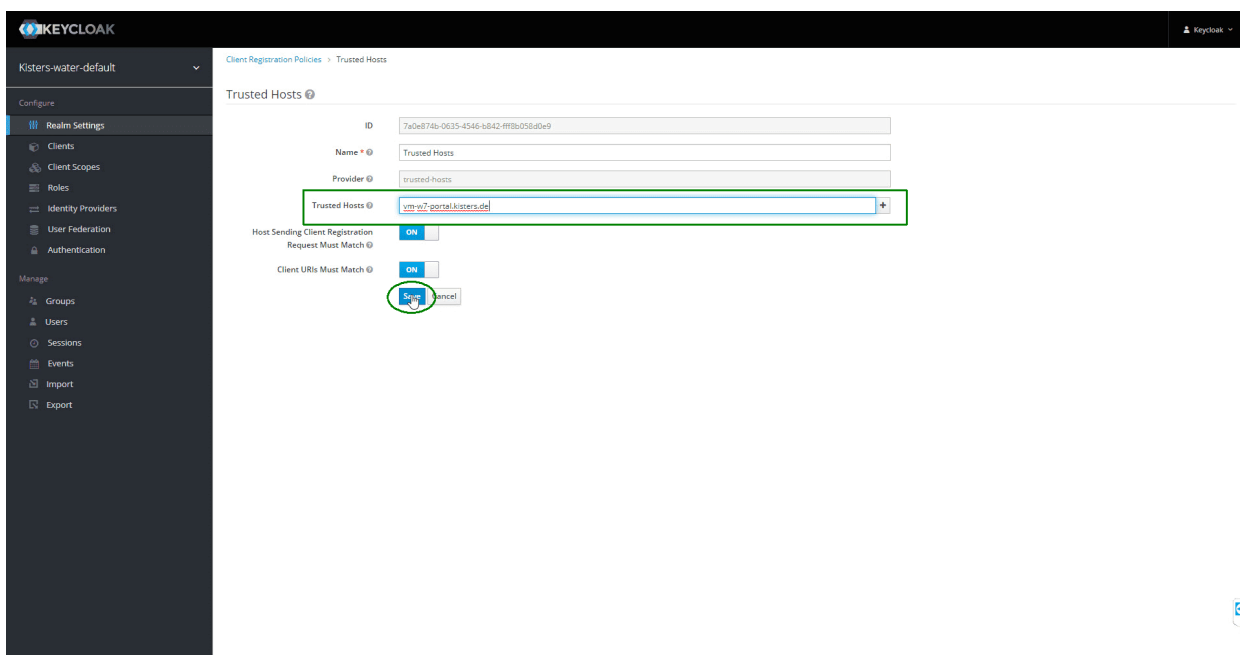
Anonymous Access Policies

Policy Name	Provider ID	Actions
Allowed Protocol Mapper Types	allowed-protocol-mappers	Edit Delete
Max Clients Limit	max-clients	Edit Delete
Consent Required	consent-required	Edit Delete
Full Scope Disabled	scope	Edit Delete
Allowed Client Scopes	allowed-client-templates	Edit Delete
Trusted Hosts	trusted-hosts	Edit Delete

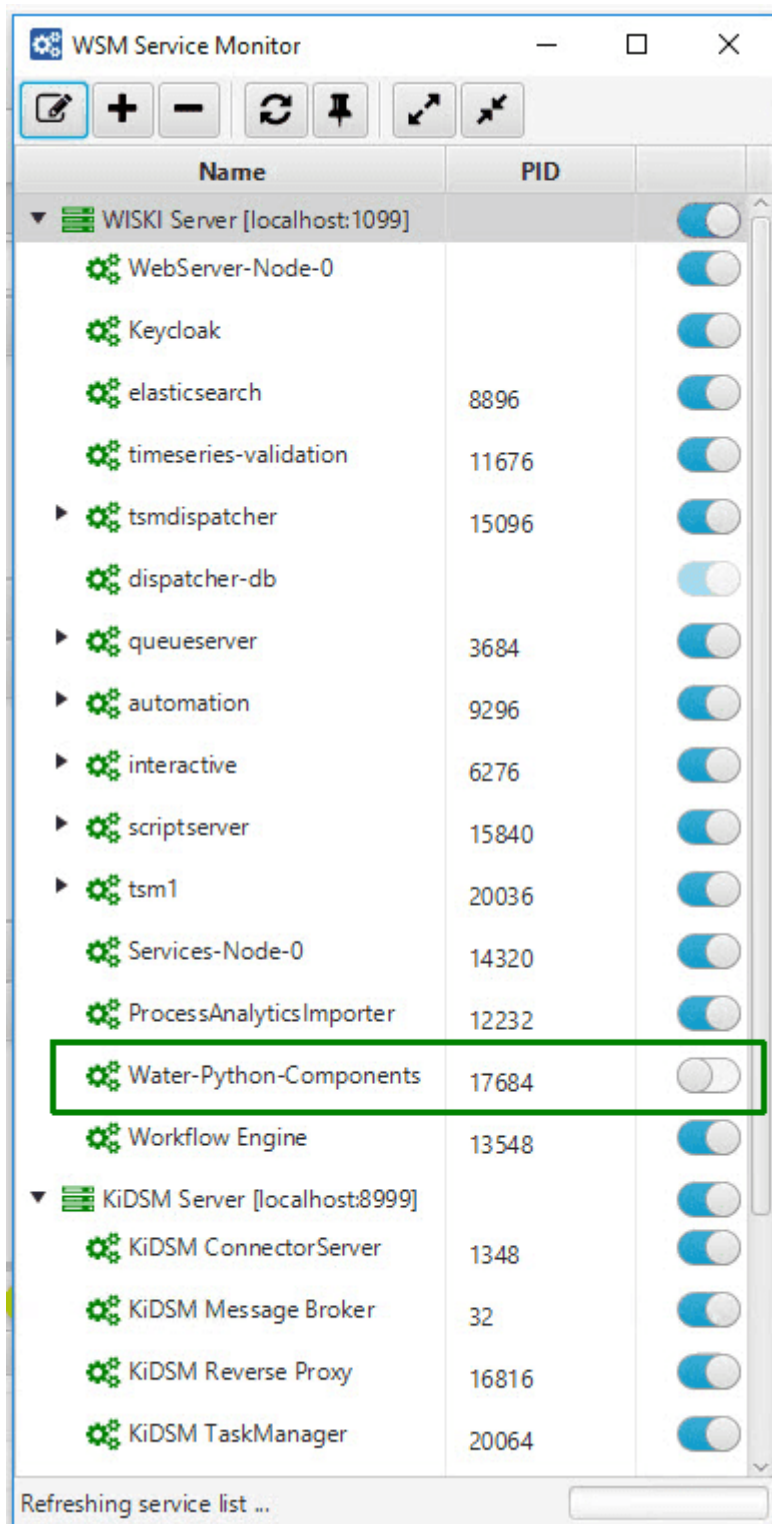
Authenticated Access Policies

Policy Name	Provider ID	Actions
Allowed Client Scopes	allowed-client-templates	Edit Delete
Allowed Protocol Mapper Types	allowed-protocol-mappers	Edit Delete

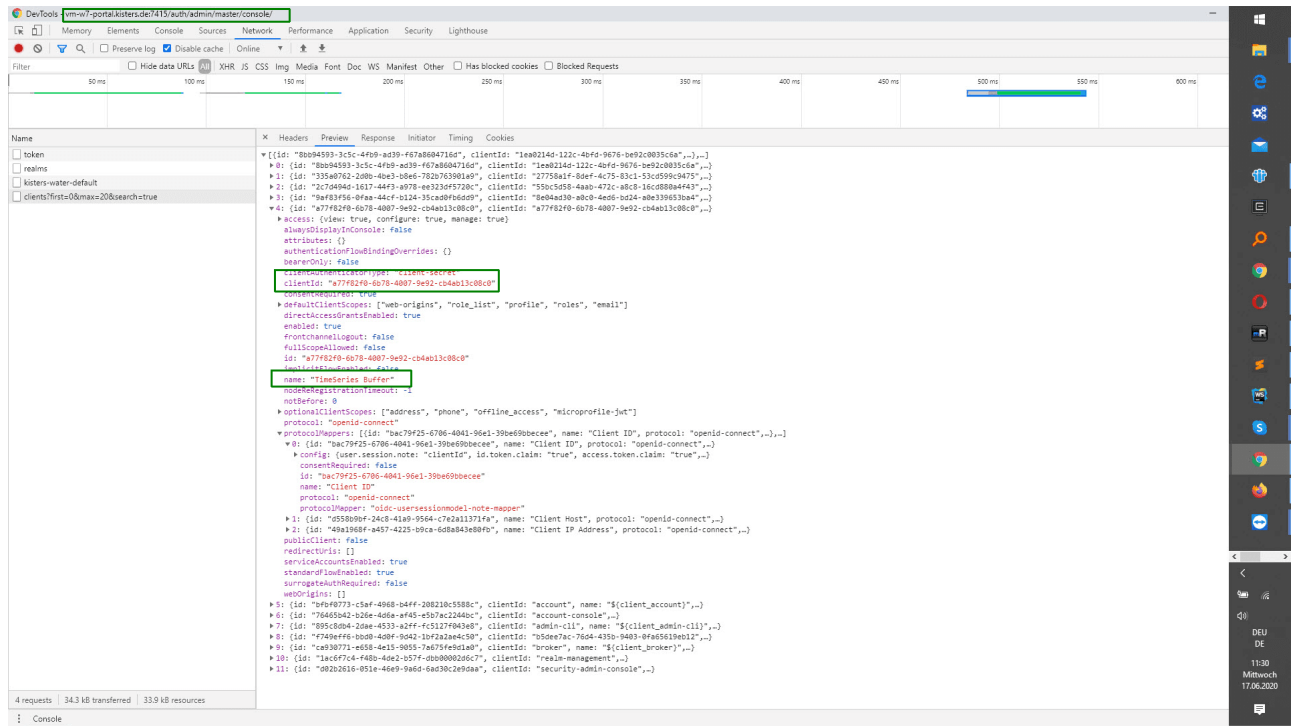
<https://vm-w7-portal.kisters.de/7415/auth/admin/master/console/#/realms/kisters-water-default/client-registration/client-reg-policies>



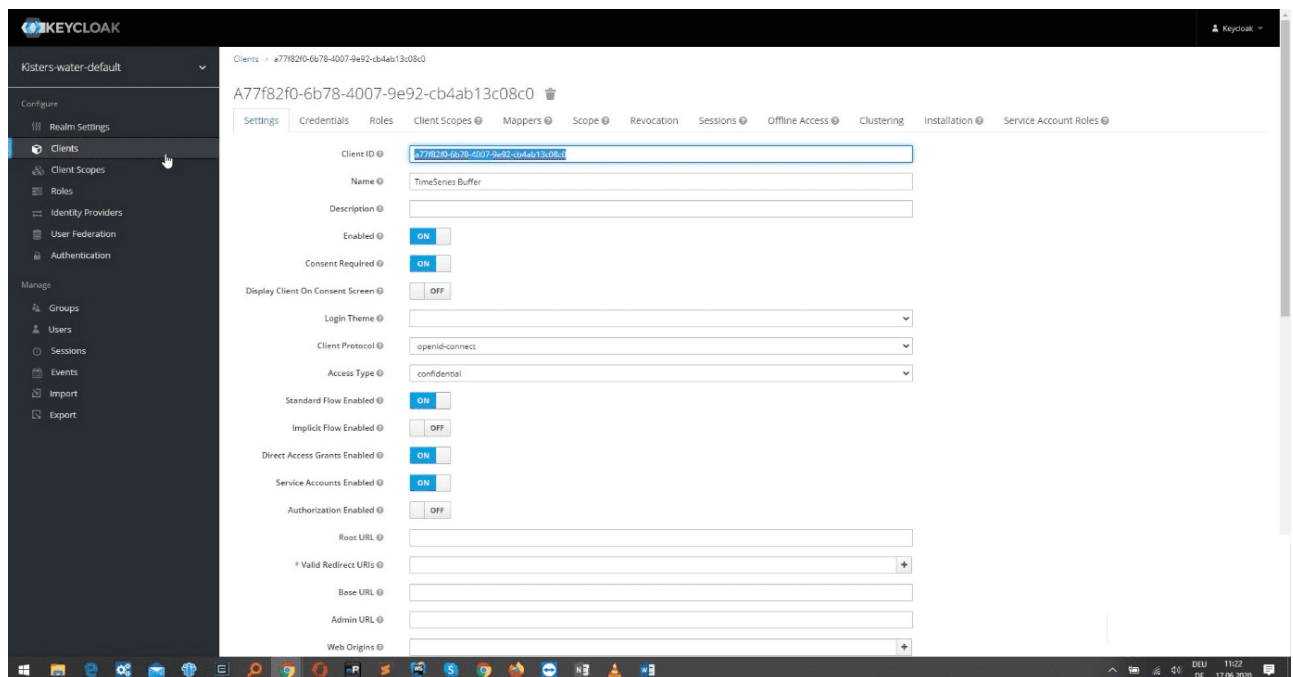
- Restart the **Water-Python-Components** node in the WISKI Service Monitor.



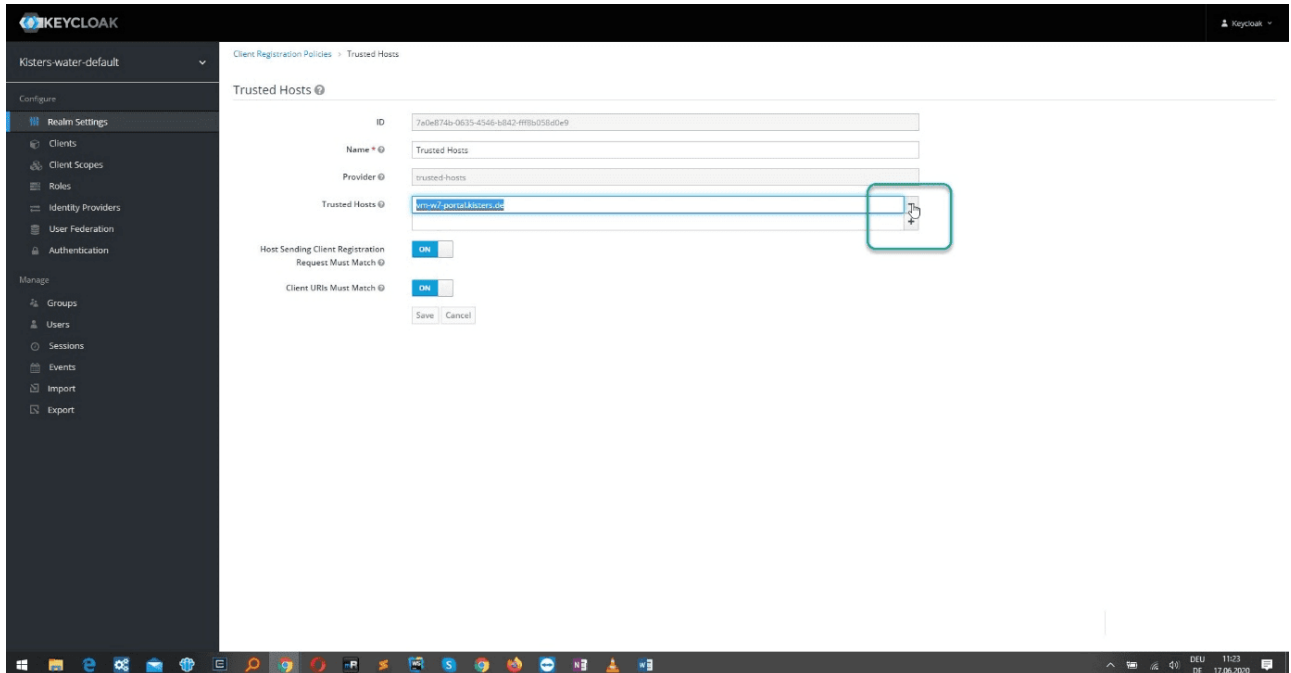
- If the **Water-Python-Components** node is running again, check whether the registration was successful.
 - Find the correct client ID (e.g. with Chrome debugger).



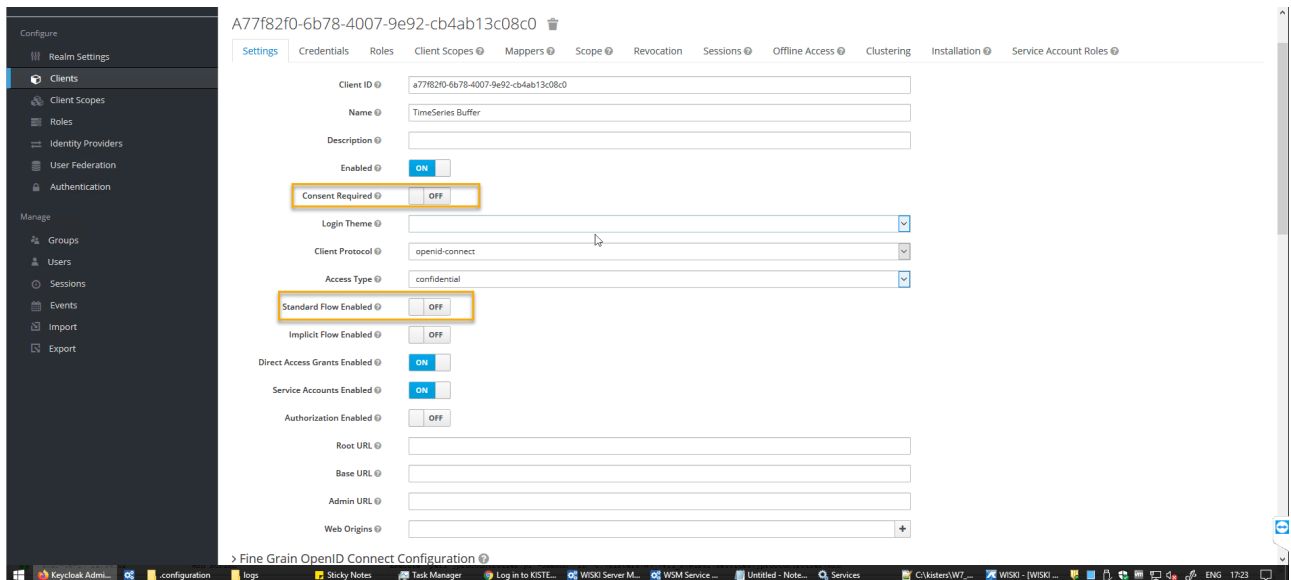
- Verify that the name of the client is registered as **TimeSeries Buffer** in Keycloak management.



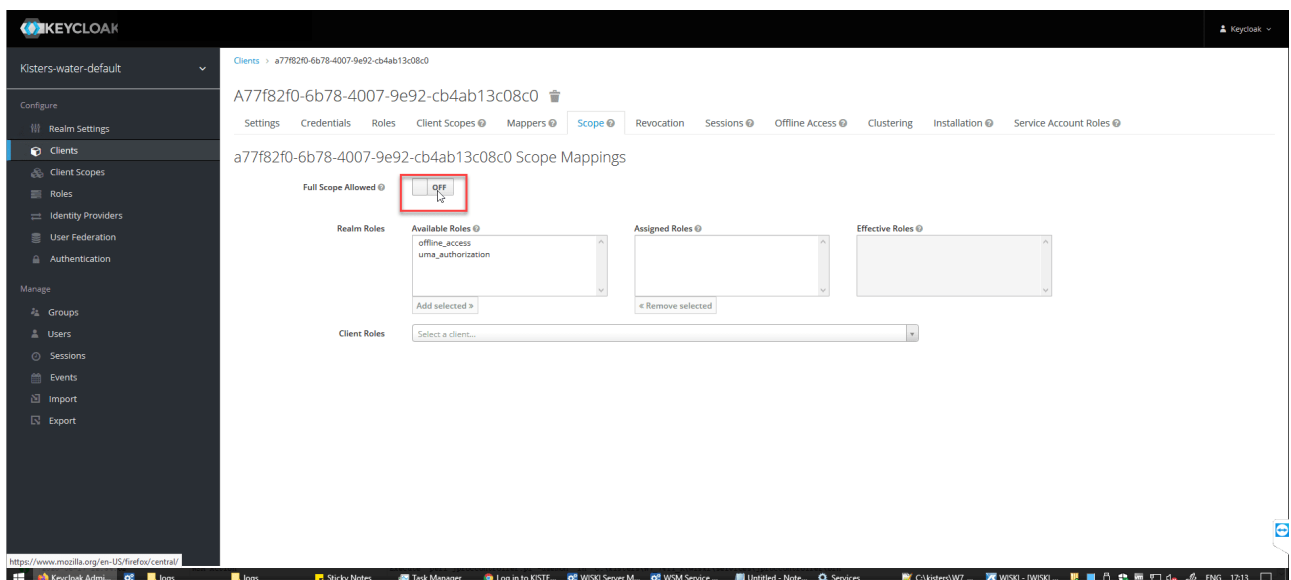
- Now you can remove your hostname from the **Trusted hosts**.



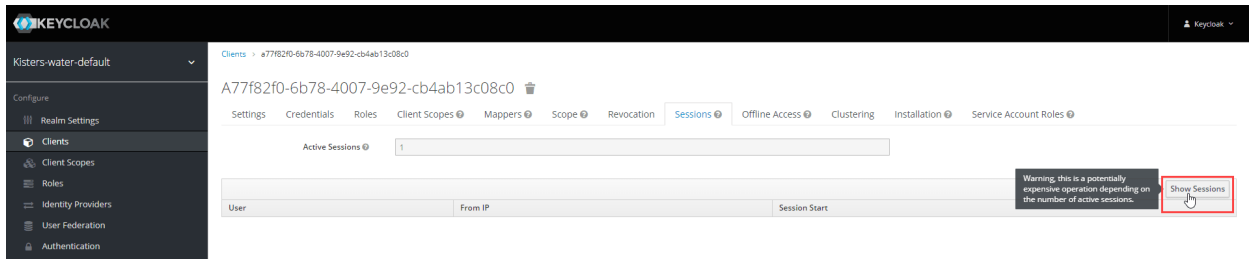
- Modify the **Settings** for the newly created client.
 - Disable **Consent Required** and **Standard Flow**.



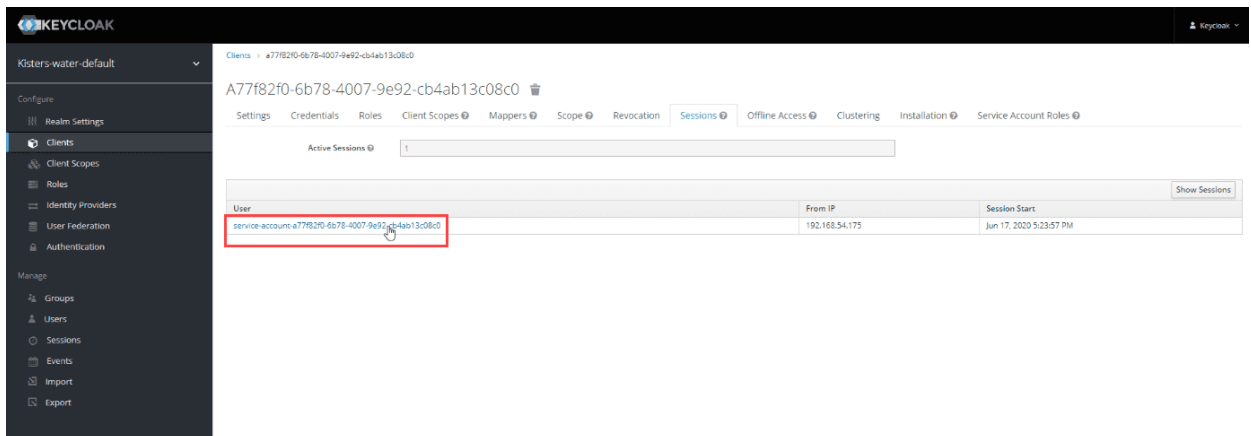
- Turn on the **Full Scope Allowed** toggle.



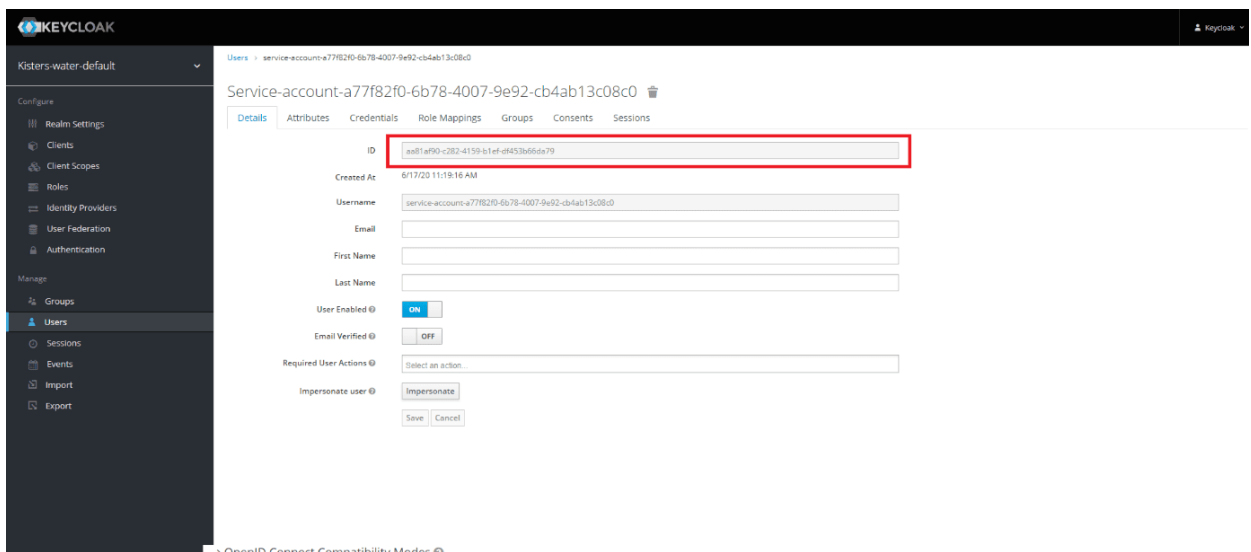
- Add the client to the WISKI database (User Administration).
 - Get the correct UUID by performing the following steps.
 - Use the interface **CLIENTS > SESSION > SHOW SESSIONS**



- Select the service account from the user list.

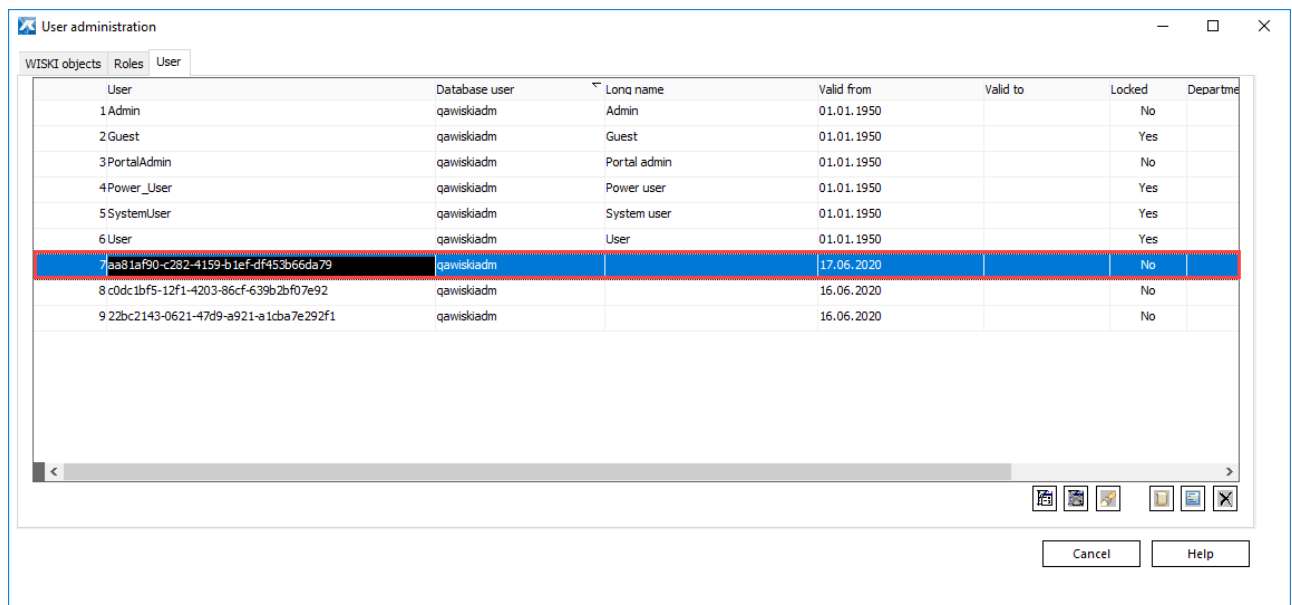


- Copy the ID to your clipboard.

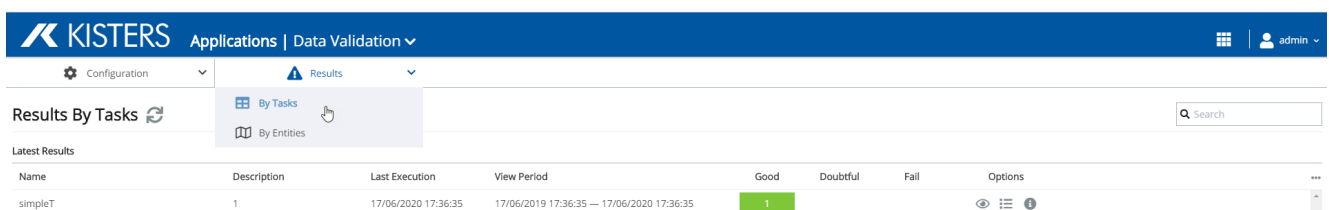


- Add the client to the list of users in your WISKI user administration.

Note: User is the ID from your clip. Grant the user administrator rights.



- The time series buffer should be running. After starting tasks the result should be visible in the interface of the web application.



18.5 Standard Applications

Currently, Application Metrics and Process Analytics are available within the standard WISKI7 version (since 7.4.7 SR14).

From WISKI 7.4.11 SR7 on, the basic version of the Data Viewer is included in each portal installation. With a licence, the KiDSM web client can be integrated in the Standard Portal since SR5.

Detailed information can be found in the next subchapters:

- [Application Metrics](#) ¹⁹⁵
- [Process Analytics](#) ¹⁹⁷
- [Data Viewer](#) ¹⁹⁹
- [KiDSM](#) ²⁰⁰
- [TS Data Validation](#) ²⁰³

18.5.1 Application Metrics

Core functionality

The Portal application Application Metrics is part of the WISKI Standard Portal. It is based on the **kigetmetrics-win-64** package, which can be selected upon installation or update using the WSM (package selection **portal** and **kigetmetrics-64-bit-win**). As one of the WISKI viewer applications, it mainly offers a web-based visualisation of the status of import log files and system performance.

See the separate user documentation for Application Metrics that is shipped with the package **portal** when being selected in the WISKI Server Manager or can be separately obtained at KISTERS. It is also available as PDF file within the application itself in the **HELP** tab.

Features

Application Metrics comes with two default views and a help tab:

View	
Metrics Overview	This view provides information on metrics considering time series marked as invalid, pending calculations and imports done in the past hour. Each view can be customized and maximized. A preselection helps to set time periods of interest.
Metrics List	In the METRICS LIST view, specific information about locations, parameters, time ranges and time series paths are shown to make a detailed selection. After choosing metrics time series, a graph view can be displayed to show the selected time series in one graph or in a matrix to compare the data.
Help	Will show the user manual as an integrated PDF file.

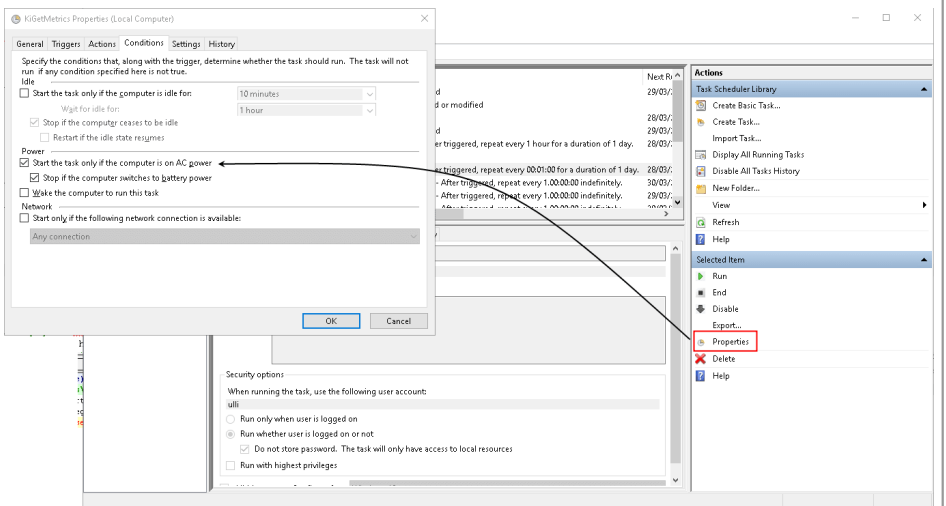
See the separate user documentation for Application Metrics that is shipped with the package **portal** when being selected in the WISKI Server Manager or can be separately obtained at KISTERS.

Installation and configuration

The configuration is based on KiGetMetrics. The WSM will take care of a default configuration on Windows systems. A scheduled job should be selected, too. On Linux systems, a Cron job can be installed.

Troubleshooting

Issue	Solution
Metrics cannot be viewed in Application Metrics.	<ul style="list-style-type: none"> Check if the package KiGetMetrics has been installed and a minutely scheduled job is running the file <code><your_WISKI_version>/tools/kigetmetrics/KiGetMetrics.bat</code> in the WISKI7 installation's folder. Check if the database connections, targets and metrics have been configured correctly in <code><your_version>\tools\kigetmetrics\config\KiGetMetrics.xml</code>. Check if the scheduled task is running correctly (see next section). Check if the exchange database temporarily holds any data (for MSSQL: <code>[dbo].[app_metricexchange]</code>; should have values until WISKI7 imports them). Examine the log file <code>(<your_version>\tools\kigetmetrics\logs\KiGetMetrics.log)</code> to make sure KiGetMetrics is writing values. Restart the server after performing modifications to relevant parts of the software.
The scheduled task is not running (Error message: "The operator or administrator has refused the request 0x800710E0").	Uncheck the power setting START THE TASK ONLY IF THE COMPUTER IS ON AC POWER :

Issue	Solution
	

18.5.2 Process Analytics

Core functionality

As a successor to the former KiLogViewer, the KISTERS Process Analytics tool aims towards providing a good overview of system logs running on different systems. Currently, that comprises KiIOSys, KiRollingOptimizer, KiDSM as well as SODA. With the new application, messages across these four components – whether they are warnings, success or error messages – can be checked at a quick glimpse within one app. Designed as a scalable web application, future components can be added according to the customer's needs.

The application itself can be reached through the WISKI Standard Portal (also known as Web Portal), the URL to call within your favourite browser is http://<server_ip>.kisters.de:7481/applications/#pca/overview. With HTTPS, you would use port 7482 by default. Alternatively, it can be accessed via the WSM by clicking on the WISKI Portal icon in the **ACTIONS** tab and switching to the application after login in your default browser.

See the separate user documentation for Process Analytics that is shipped with the package **portal** when being selected in the WISKI Server Manager or can be separately obtained at KISTERS. It is also available as PDF file within the application itself in the **HELP** tab.

Features

Process Analytics comes with these default views:

View	
Overview	Provides the log overview comprising four components, i.e. SODA, KiRollingOptimizer, KiDSM and KiIOSys. Further systems can be added in future, according to customer's needs. A process chain will provide information about the components involved, e.g. to keep track of which import has yielded an error. Users thus no longer have to find and check single log files across their file system.
Messages	This view lists all the messages across components. Filters are available to narrow down your search. For performance reasons and to keep an overview, the list of messages is restricted to 5000 rows – use the filters to display only what you need. Amongst others, use cases might be detecting log messages like TIME SERIES NOT FOUND or issues when exchange numbers could not be processed adequately.
Daily Overview	To quickly see what happened on a selected day, the user will switch to this view. The full day is shown in a clickable ribbon with the number of messages written each

View	
	hour. For a customised view, individual columns with information about the day's processes can be displayed or hidden.
Help	Will show the user manual as an integrated PDF file.

Installation and Configuration

In order to get Process Analytics running, you must install three packages: WISKI Standard Portal, the Elasticsearch Module (for the No-SQL database) and the Process-Analytics Importer. Select the additional packages (**portal**, **elasticsearch**, **log-processanalytics**) when installing or updating via the WSM, which will automatically install and configure your system properly. The application can be stopped or started conveniently using the Service Monitor of the WSM or the JProcController Console individually, which will not close any other services or applications. Remember that the Process Analytics Importer always needs the Elasticsearch database.

For advanced users, the tool can be managed via CLI, too. With a Kibana installation, advanced users can also access the Elasticsearch database itself.

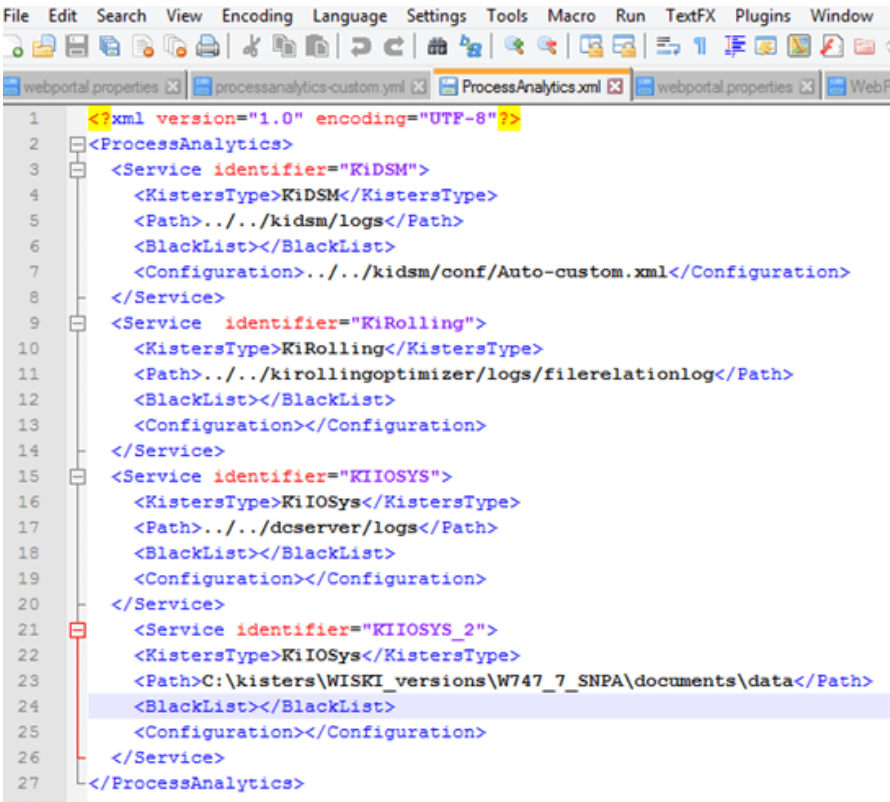
Main configuration files are:
<code>...wiski\services\processanalytics\conf\ProcessAnalytics.xml</code>
<code>...wiski\services\processanalytics\conf\processanalytics-custom.yml</code>

The maximum memory can be set (from importer version 1.8.0 / 7.4.11 SR15) as follows:

- Edit the file `...wiski\services\processanalytics\conf\jvm.options`
 - Adjust the `-Xms` and `-Xmx` entries to the desired values. The examples in the file give some hints which values might make sense, for normal systems the default 1 GB should be fine.
- Note:** If the WISKI server is running, the ProcessAnalyticsImporter node must be restarted to use the new settings.

Troubleshooting

Issue	Solution
If you do not see any data in the WISKI Standard Portal viewer, ...	<ul style="list-style-type: none"> Check if the Process Analytics importer is really running. Its node should have a PID in the Service Monitor of the WSM as well as in the JProcController console. If it is not running, you can restart it manually by activating the node or with the batch script <code>importer_start.bat</code> found in <code>...wiski/services/processanalytics</code>. Check if the <code>importer_status.bat</code> yields "OK" status for all instances. If not, try restarting as described above. This file will also show the PID, which can be killed via CLI, too.
If you lack information on components / processes in your Process Analytics Viewer, ...	<ul style="list-style-type: none"> Check if you have registered the corresponding service and it is not black-listed in your <code>ProcessAnalytics.xml</code> file:

Issue	Solution
	 <pre> 1 <?xml version="1.0" encoding="UTF-8"?> 2 <ProcessAnalytics> 3 <Service identifier="KiDSM"> 4 <KistersType>KiDSM</KistersType> 5 <Path>../../kidsm/logs</Path> 6 <BlackList></BlackList> 7 <Configuration>../../kidsm/conf/Auto-custom.xml</Configuration> 8 </Service> 9 <Service identifier="KiRolling"> 10 <KistersType>KiRolling</KistersType> 11 <Path>../../kirollingoptimizer/logs/filerelationlog</Path> 12 <BlackList></BlackList> 13 <Configuration></Configuration> 14 </Service> 15 <Service identifier="KiIOSYS"> 16 <KistersType>KiIOSys</KistersType> 17 <Path>../../dcserver/logs</Path> 18 <BlackList></BlackList> 19 <Configuration></Configuration> 20 </Service> 21 <Service identifier="KiIOSYS_2"> 22 <KistersType>KiIOSys</KistersType> 23 <Path>C:\kisters\WISKI_versions\W747_7_SNPA\documents\data</Path> 24 <BlackList></BlackList> 25 <Configuration></Configuration> 26 </Service> 27 </ProcessAnalytics> </pre> <ul style="list-style-type: none"> ▪ The Process Analytics Importer will run as a java.exe process traceable in the Windows Task Manager – showing the details with the command line will show the importer process, which can be killed from here, too. ▪ For further troubleshooting, check the log file located at: ... wiski/services/processanalytics/processanalytics_importer.bat.

18.5.3 Data Viewer

Core functionality

The basic Data Viewer is a new application which is shipped with the WISKI7 Standard Portal from WISKI 7.4.11 SR7 onwards. The Data Viewer is included in the standard package of the WISKI Server Manager and is automatically installed when the package portal is selected and does not require complex configurations in WISKI or the Data Viewer itself. It shows WISKI content in a web GUI with read-only access and requires access to WISKI to display your stations and corresponding time series in real time. In addition to real-time data, you also have access to historical data.

This free version provides you a standard viewer. It is a basic version of the more complex Water Data Viewer. The Water Data Viewer can be configured specifically to your needs and requirements. In case you look for a highly configurable internal web page, please contact your KISTERS project representative for more information about our products.

Features

The Data Viewer shows WISKI content for readonly access and presents users with these particular benefits:

- Mapping using background maps from ESRI ArcGIS Online
- Enhanced time series graphing with interactive graphs
- Download features for data in XLSX, CSV other formats such as WaterML2
- Interactive selection of time series from various stations for graphing and for downloads
- Station overview either as map layer over a map or in a configurable list
- Searching and filtering: within the station overview; stations can be filtered according to station metadata fields or with a full text search

Installation and Configuration

To install the Data Viewer, select the **PORTAL** package in the WISKI Server Manager. **KIWIS** and **WDP** are required as well (select the packages **kiwis** and **wdp-win** in the WISKI Server Manager wizard dialog).

Troubleshooting

Issue	Solution
If you do not see any location markers in your map (a white map area), ...	<p>Make sure that the global coordinate columns in your station location tab is filled. Check your WISKI stations, they must contain position information, too. You can use this web page to obtain or transform appropriate coordinates for your area: Coordinate Systems Worldwide. WISKI7 uses WGS84 with EPSG:4326 as global SRS (spatial reference system).</p> <p>See the comprehensive information in the WISKI TCA client user guide or online help under "General Concepts" > "Maps and Spatial Referencing".</p>

18.5.4 KiDSM

Core functionality

Since WISKI 7.4.11 SR7, the former RCP client is no longer shipped. Users must use the KiDSM application which is integrated in the Standard Portal.

KiDSM automates many different steps in your daily work with KISTERS programs. Use the program e.g. for automatic basic data and time series import and export, for data forwarding in the Power context and for automatic report generation in all KISTERS programs. You can monitor files and locations (directories), convert files, download data from web or FTP sites, trigger alarms and much more. All these operations can be started immediately or according to a schedule.

Features

- Dashboard: most important information at first sight
- Configuration Overview: configuring tasks, locations and jobs to get started
- Tasks Overview: the current status of all your tasks (progress, valid from/until, last execution)
- Jobs Overview: the current status of all your jobs (activity, progress, locations)
- Key Management: managing keys (e.g. tls-rest-key pairs)
- Process Overview: the current status of all your processes (retries, latency and more)
- Jobs Statistics: statistical overview with execution, transmission, file detection, successful or failed state information
- Notification View: configuration of notification and receivers
- Clean-Up View: to schedule the deletion of log files and clean up KISTERS components
- Import Configuration: for xml-based import configurations
- Export Configuration: for xml-based export configurations
- Job Diagram: a visualisation of components being involved in jobs, tasks and groups

Installation and Configuration

For WISKI versions 7.4.11. SR5 and SR6, refer to the troubleshooting section below. Since the release of WISKI 7.4.11 SR7, by default, only the WISKI Admin user (with the role **UserAdmin**) can see the integrated KiDSM client. The Admin user can assign permissions to each user who is to have access to this application (see [Portal Administration](#)^[185], "User settings" section).

Troubleshooting

Issue	Solution
For versions 7.4.11 SR5 and SR6, the permissions must be modified manually. The following steps describe how to proceed in detail.	see below

Log in at the WISKI Standard Portal with user **PortalAdmin** and select the application **PORTAL ADMIN**.

Check the portal object permission groups:

KISTERS Admin Portal Admin					
Portal Object					
Portal Objekte Berechtigungsgruppen				Export	Suchen
Key	Beschreibung	Erstellungsdatum	Änderungsdatum	Optionen	
PO KIDSM	KIDSM Permissions	20.12.2019 16:48	20.12.2019 16:48	[Edit] [Copy] [Delete]	
PO Metrics	Metrics Portal Object Permissions	05.11.2018 11:18	05.11.2018 11:18	[Edit] [Copy] [Delete]	
PO ProcessAnalytics	ProcessAnalytics Portal Object Permissions	05.11.2018 11:18	05.11.2018 11:18	[Edit] [Copy] [Delete]	
PO SystemAdmin	All Portal Object Permissions	05.11.2018 11:18	28.10.2019 14:30	[Edit] [Copy] [Delete]	
PO Wiski User	Wiski User Portal Object Permissions	05.11.2018 11:18	05.11.2018 11:18	[Edit] [Copy] [Delete]	

Create the portal object **KIDSM** with the description **KIDSM Permissions** if it does not exist:

Portal Object

[Zurück](#)

PO KIDSM

KIDSM Permissions

Portal Objekt Berechtigungen

Suchen

Typ	Name	Level
APPLICATION	Portal Admin	Keine Lesen Schreiben Löschen
APPLICATION	Userspace	Keine Lesen Schreiben Löschen
APPLICATION	Application Metrics	Keine Lesen Schreiben Löschen
APPLICATION	KIDSM	Keine Lesen Schreiben Löschen
APPLICATION	Process Analytics	Keine Lesen Schreiben Löschen

Erstellen

Check the user roles:

Issue
Solution

KISTERS Admin | Portal Admin
PortalAdmin

- Users
- Users & Roles
- Roles**
- Permissions
- Functional
- Portal Object
- Data Source
- Applications
- Packages
- Portal Settings
- WebLogs
- Information
- Widget Class
- Announcement

Roles
Back

KIDSM

KIDSM User Role

Permission groups

Active	Key	Type	Description	Created	Edited	Options
☐	portal-client-userspace.all	FPG	all permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	portal-client-admin.menu.userAdmin	FPG	Edit only users	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	F SystemAdmin	FPG	All Functional Permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	portal-client-core.all	FPG	all permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	F ContentManager	FPG	Content Management Functional Permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	portal-client-admin.menu.all	FPG	Edit everything	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	F Dashboards	FPG	Dashboard Functional Permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	F ProcessAnalytics	FPG	ProcessAnalytics Functional Permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	F Wiski User	FPG	Wiski User Functional Permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	PO ProcessAnalytics	POPG	ProcessAnalytics Portal Object Permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	PO SystemAdmin	POPG	All Portal Object Permissions	05/16/2019 9:01 AM	12/17/2019 11:19 AM	
☒	PO KIDSM	POPG	KIDSM Permissions	12/20/2019 5:23 PM	12/20/2019 5:24 PM	
☐	PO Wiski User	POPG	Wiski User Portal Object Permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	
☐	PO Metrics	POPG	Metrics Portal Object Permissions	05/16/2019 9:01 AM	05/16/2019 9:01 AM	

Create the user role **KIDSM** with the description **KIDSM User Role**, if it does not exist.

Roles
+ Add

Name	Description	Role Level	Is Default
ContentManager	Content Manager Role	7	
KIDSM	KIDSM User Role	0	
Metrics	Metrics User Role	0	✓
ProcessAnalytics	Process Analytics User Role	0	✓
SystemAdmin	System Administrator Role	11	
WiskiUser	Wiski User Role	0	✓

Assign the role **KIDSM** to the user **Admin**:

KISTERS Admin | Portal Admin
PortalAdmin

- Benutzer
- Users & Roles
- Benutzerrollen
- Permission & Roles
- Berechtigungen
- Funktionale
- Portal Objekte
- Datenquellen
- Applikationen
- Pakete
- Portal Einstellungen

Users
Zurück

Grundeinstellungen Benutzerrollen Benutzerdaten Einstellungen

Roles

Aktiv	Benutzerrolle	Beschreibung	Options
☐	ContentManager	Content Manager Role	
☒	WiskiUser	Wiski User Role	
☒	ProcessAnalytics	Process Analytics User Role	
☒	Metrics	Metrics User Role	
☐	SystemAdmin	System Administrator Role	
☒	KIDSM	KIDSM User Role	

When you install for the first time, you might get the error **KIDSMURL NOT FOUND** in the web GUI.

If that occurs, check the configuration file

`\wiski\services\tomcat\node0\webapps\applications\portal-client-kidsm\package.json`.

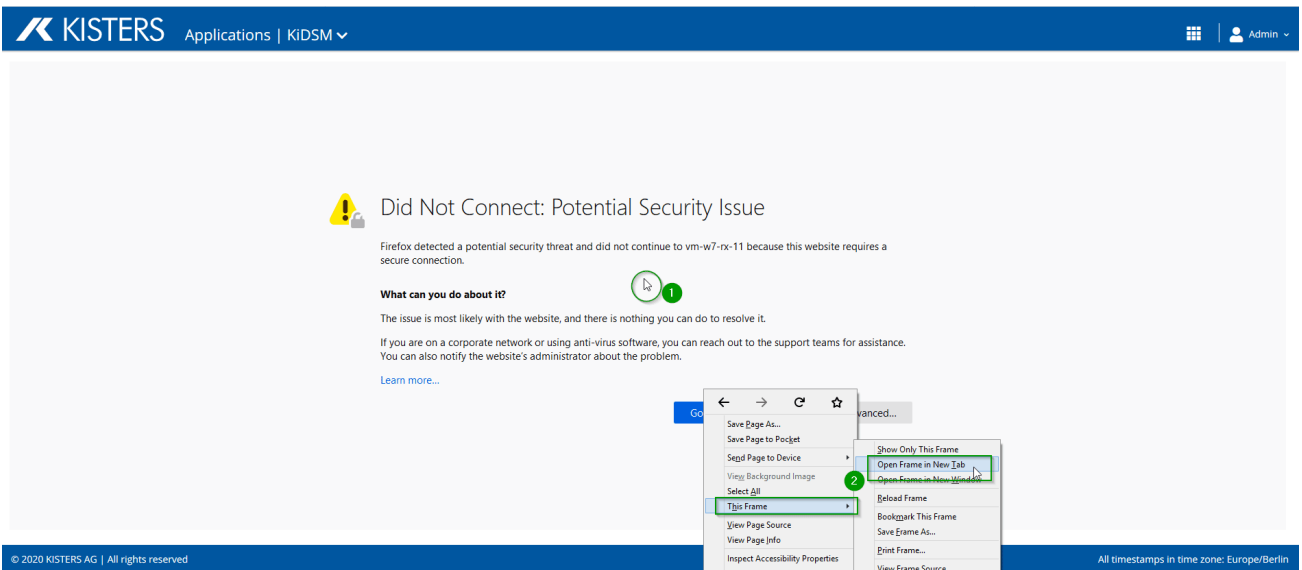
If the `kidsmURL` value is empty, replace the "" with the correct URL

`https://<hostname>:8843`

Issue	Solution
	The port should be 8843 and the host name and protocol (http or https) must be in line with your portal configuration (compare <code>\wiski\services\portal\config\webportal.properties</code>).
When using Firefox and only self-signed certificates, opening the KiDSM client integrated into the WISKI standard portal indicates a "Potential Security Problem" that is not critical.	see below

If you do not want to use Chrome as the recommended browser, you must first open the standalone KiDSM in your Firefox browser session and then switch back to the integrated client:

- Right click in the iframe window and select **THIS FRAME** and then **OPEN FRAME IN NEW TAB**.



- Switch to the newly opened tab, click on **ADVANCED...** and **ACCEPT RISK AND CONTINUE**.
- Switch back to the WISKI Standard Portal tab and reload the portal (e.g. by pressing F5). You can close the other tab (with the standalone client) and continue working with the integrated KiDSM web client.

18.5.5 TS Data Validation

Core functionality

Your everyday tasks rely on complete and validated data. This application helps you to notice any inconsistencies and to initiate appropriate actions. You can configure individual validation check types and define time series groups to be used for the validation checks. Checks can be scheduled, results are stored in the corresponding backend (e.g. the WISKI database or in other attached information systems or programs). The application is useful for both high resolution and long term information (e.g. before preparing annual reports).

TS Data Validation provides you with a Web-based user interface for the administration and visualisation of data validation rules and tasks. You can define and manage these rules and visualise the results of the validation runs on maps and in tables. Using workflows, you can manage the scheduled execution of the tasks. With this, your time series data can be presented in a viewer application.

The application is part of the WISKI Standard Portal.

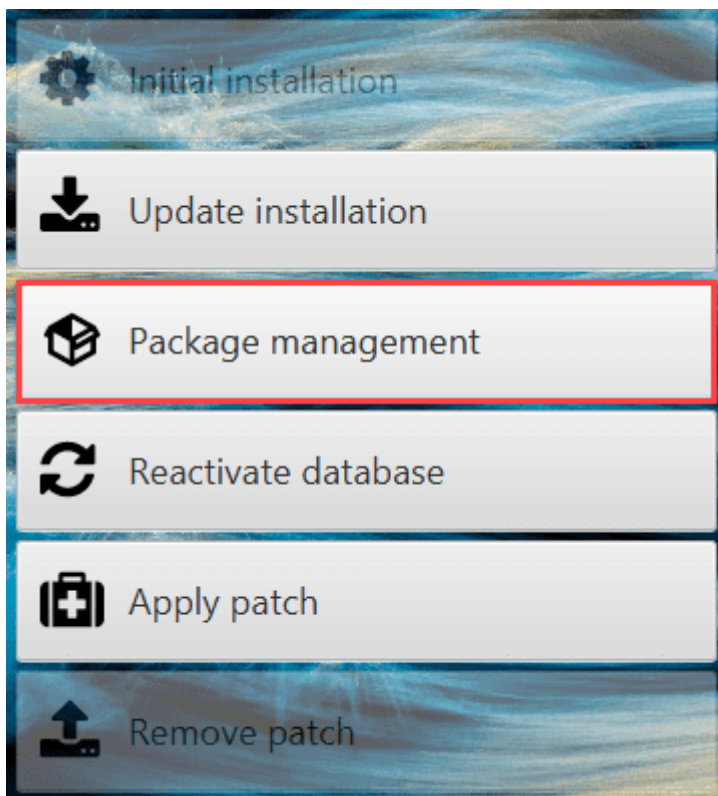
Features

- Defining rules is the first of several steps in configuring validation routines. The different types of rules to be configured are directly linked to the validation algorithms assigned in the server side of the system.

- Task definition is the next step after rule configuration. In this step, a link is created between a task and existing rules, and a validation execution can be performed based on this link.
- Workflow definition is the next step after rule configuration and task generation. In the Data Validation application, a workflow is the scheduled execution of tasks. This functionality allows you to set an execution interval for one or more previously configured tasks.
- The **RESULTS** view consists of the Results Overview, Results Detail and Time Series views. The information from completed task instances can be analysed and evaluated here.
- The Results Overview displays the results either by Task or by Entity, the latter representing the Latest Results table in the Results Summary by Task.

Installation and Configuration

With a valid license, you will have to install Python and some further backend services. A customized Python will be shipped with your WISKI version (Miniconda, about 1.2 GB). Select the following packages in the **WISKI SERVER MANAGER > PACKAGE MANAGEMENT**:



Note: The following two packages must be installed in the specified order.

- **water-python**
- **python-jproc**

The following three packages can be installed in any order.

- **validation-karaf**
- **validation-python**
- **water-workflow-engine**

The default configuration which is suggested by the WISKI Server Manager during installation will work for most installations.

Mind that during the installation process, you might be prompted to install Python locally and your firewall might ask for permission – you will have to allow communication to continue.

After a successful initial installation, only the **PortalAdmin** user can see the application. That user can assign permissions to each user who should have access to this application (see [Portal Administration](#)^[185], **USER SETTINGS** section).

If you want to install the Data Validation application on a running WISKI standard portal without initiating a full WISKI update, you can also select the five packages mentioned above in the order given above (WISKI Server Manager: **PACKAGE MANAGEMENT > INSTALL PACKAGE**). If necessary, standard configurations are also offered in the installation dialogs.

Note: If you are using WISKI 7.4.11 SR8 or SR9 with Keycloak, refer to the [Troubleshooting for the Portal – Versions with Data Validation \(7.4.11 SR8 and higher\)](#)^[186] section.

Troubleshooting

Issue	Solution
<p>If you do not receive any data when you select the application in the GUI and a red pop-up window with the message ERROR LOADING DATA appears, ...</p>	<ul style="list-style-type: none"> Make sure that all services are registered in the service registration. Visiting https://<hostname>:7421/serviceregistry must show that the following services are running for Data Validation: <ul style="list-style-type: none"> DE.KISTERS.TIMESERIES.BUFFER DE.KISTERS.TIMESERIES.VALIDATION DE.KISTERS.TIMESERIES.VALIDATION.REPOSITORY DE.KISTERS.TIMESERIES.VALIDATION.RULES.IMPLEMENTATION DE.KISTERS.WATER.AUTH.OPENIDCONNECT DE.KISTERS.WATER.EXECUTIONQUEUE DE.KISTERS.WATER.WORKFLOWENGINE
<p>If you cannot edit time series but get the error message WISKIDATAEXCEPTION: MISSING PERMISSION TSM.TSMSEVER.CHECKOUTTSDAT AWISKIDATAEXCEPTION: MISSING PERMISSION TSM.TSMSEVER.CHECKOUTTSDAT A, ...</p>	<ul style="list-style-type: none"> Check the user's authorisations for the time series buffer in WISKI. Check whether the user can be identified via the UUID in the WISKI user administration. <ul style="list-style-type: none"> This UUID must be the same as in your OIDC provider (MITREid: MANAGE CLIENTS > TIMESERIES BUFFER: Edit CLIENT > MAIN: Client ID) – for Keycloak see Troubleshooting for the Portal - Versions with Data Validation (7.4.11 SR8 and higher)¹⁸⁸. Test by granting Admin rights in the WISKI Client User Administration and do the editing again.
<p>If tasks and workflows are not launched properly, ...</p>	<ul style="list-style-type: none"> Check the time series buffer user rights as in the step above. Restart the Water-Python-Components node in the WISKI Server Manager.

19 Appendix

The Appendix collects some miscellaneous information:

- [External Java installations](#) ^[206]
- [Windows Services](#) ^[206]
- [Client Installation Using an Installation Wizard \(WiX Toolset\)](#) ^[206]
- [Client Installation/Update in Silent Mode](#) ^[220]
- [Usage of Elasticsearch](#) ^[223]
- [Python as Scripting Language](#) ^[227]
- [FAQ](#) ^[228]

19.1 External Java installations

The WISKI Server installation needs a Java JDK installation. The JDK is delivered, installed and updated together with the other packages. Beside the normal installation it is possible to install your own Java. By installing another Java, the normal installation routines can not be used any more, they are designed to use the delivered Java Version. The WISKI Server software has to be set up manually. The latest WISKI installation always brings the newest Java version.

The WISKI Server software needs:

- Java 11 (AdoptOpenJDK is recommended)
- The JDK ([Java SE Development Kit](#)) version
- The JCE (**Java Cryptography Extension**) package installed, this can be downloaded from Oracle, please follow the installation instructions.
- The WISKI server is started with perl startup scripts; these scripts search for the installed Java in the following order.
 - Java inside the WISKI service directory (**services/java**).
 - Java installed with the environment variable **TSM_JAVA_HOME**.
 - Java installed with the environment variable **JAVA_HOME**.

19.2 Windows Services

You can register some WISKI elements as WINDOWS services:

- [Managing a KiDSM Service As a WINDOWS service](#) ^[206]
- [Managing the KISTERS WISKI Service as a WINDOWS Service](#) ^[207]

19.2.1 Managing a KiDSM Service As a WINDOWS service

Managing a KiDSM service as a WINDOWS service comprises these steps:

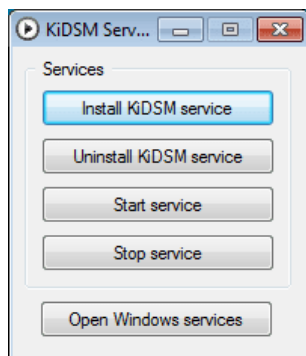
- [Launching the tool](#) ^[206]
- [Installing the WINDOWS Service](#) ^[207]
- [Uninstalling the WINDOWS service](#) ^[207]
- [Checking the service installation](#) ^[207]

19.2.1.1 Launching the tool

There are two ways to start the KiDSM Service management tool:

- Launching the KIDSM Service management tool from the WISKI Server Manager
 - Navigate to the **ACTION** tab. Double click **SERVICE - MANAGE KIDSM WINDOWS SERVICES**.
- Launching the KIDSM Service management tool manually.
 - Navigate to `<root>\wiski\bin` using a WINDOWS Explorer and execute **KiDSM-Services.exe**.

After launching the tool WINDOWS will ask for administrator rights, these have to be granted.



19.2.1.2 Installing the WINDOWS Service

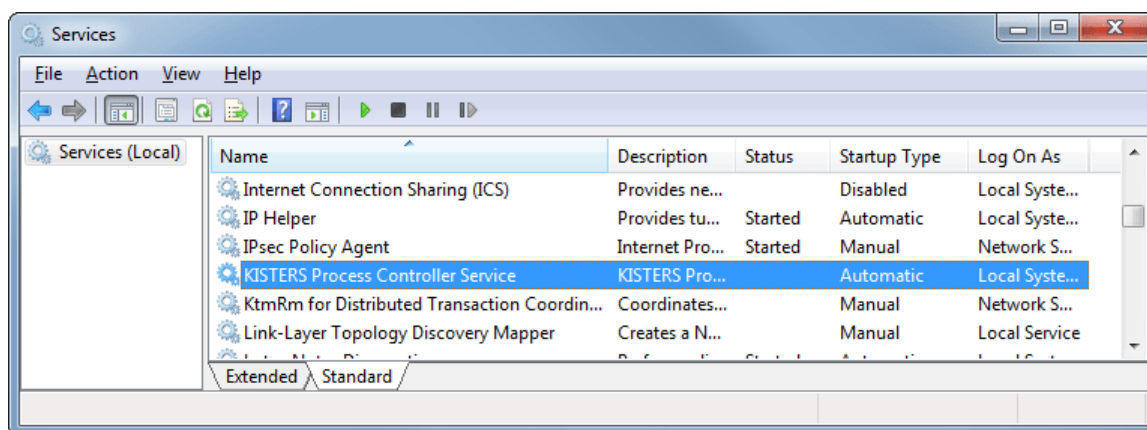
- Click the button [INSTALL KIDSM SERVICE].
- Click the button [START SERVICE].

19.2.1.3 Uninstalling the WINDOWS service

- Click the button [STOP SERVICE].
- Click the button [UNINSTALL SERVICE].

19.2.1.4 Checking the service installation

Use the button [OPEN WINDOWS SERVICES] to check if the service has successfully been installed or uninstalled.



19.2.2 Managing the KISTERS WISKI Service as a WINDOWS Service

Managing the KISTERS WISKI service as WINDOWS service comprises these steps:

- [Launching the tool](#)²⁰⁷
- [Installing the WINDOWS service](#)²⁰⁸
- [Uninstalling the WINDOWS service](#)²⁰⁸
- [Checking the service installation](#)²⁰⁸

19.2.2.1 Launching the tool

There are two ways to start the KISTERS WISKI Service management tool:

- Launching the KISTERS WISKI Service management tool from the WISKI Server Manager.
 - To shut down and uninstall navigate to the **ACTION** tab. Double click **SERVICE - MANAGE JPROCCONTROLLER WINDOWS SERVICES**.
- Launching the KISTERS WISKI Service management tool manually.
 - Navigate to <root>\wiski\bin using a WINDOWS Explorer and execute jproccontroller_services.exe.

19.2.2.2 Installing the WINDOWS service

- Click the button **[INSTALL SERVICE]**.
- Click the button **[START SERVICE]**.

Use the button **[OPEN WINDOWS SERVICES]** to check if the KISTERS WISKI service has been installed successfully.

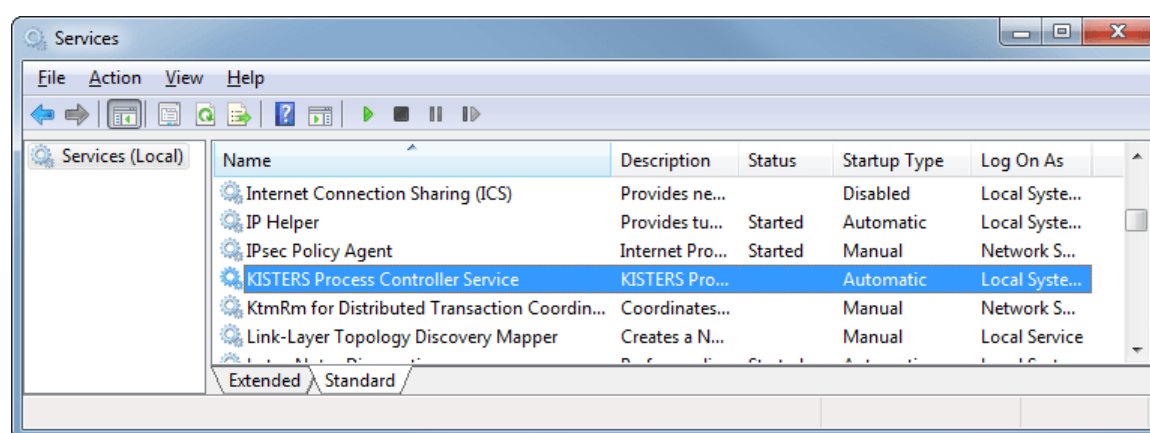
19.2.2.3 Uninstalling the WINDOWS service

- Click the button **[STOP SERVICE]**.
- Click the button **[UNINSTALL SERVICE]**.

Use the button **[OPEN WINDOWS SERVICES]** to check if the KISTERS WISKI service has successfully been uninstalled.

19.2.2.4 Checking the service installation

Use the button **[OPEN WINDOWS SERVICES]** to check if the service has successfully been installed or uninstalled.



Note: The KISTERS WISKI Services are installed under a local system account. It is recommended to use a different user. This can be changed by editing the properties of the WINDOWS service. The user itself must be created by a system administrator.

If this service has been started under the system account, a temporary folder will be added that solves problems during import of ZRXP files using the IO machine. These problems occur when writing temporary files under newer versions of the Microsoft Windows operating system. If you encounter problems when importing files or generate reports, make sure that this folder exists:

Windows	Local system user temp directory
64-bit system	C:\Windows\system32\config\systemprofile\AppData\Local\Temp
32-bit system	C:\Users\<Windows user name>\AppData\Local\Temp

19.3 Client Installation Using an Installation Wizard (WiX Toolset)

Your WISKI7 Service Monitor (as an alternative to the JProcConsole), KiDSM, KiScript and the WISKI7 client can be installed and updated together with the server using the WISKI7 Server Manager (WSM). This is [explained above](#). Apart from that a standalone MSI installer is available. This chapter describes how to install WISKI and components using this MSI installer.

Note:

- If you are updating from WISKI7 versions older than 7.4.13 SR7, uninstall WISKI7 first and then reinstall WISKI7 completely. The installation packages before version 7.4.13 SR7 were created with InstallShield.

With executing the following MSI files, you can run a client installation/update with the help of the (WiX Toolset) wizard:

- WISKI7 client
- KiDSM-Client (until 7.4.11 SR5)
- KiScriptServer IDE
- Service Monitor

You can start the wizard by double-clicking the MSI file or by running the command line from the location of the MSI file, as follows.

```
>msiexec /i <client>.msi [/l*v "<installation/update log file>"]
```

Detailed information is provided in the following sub-chapters:

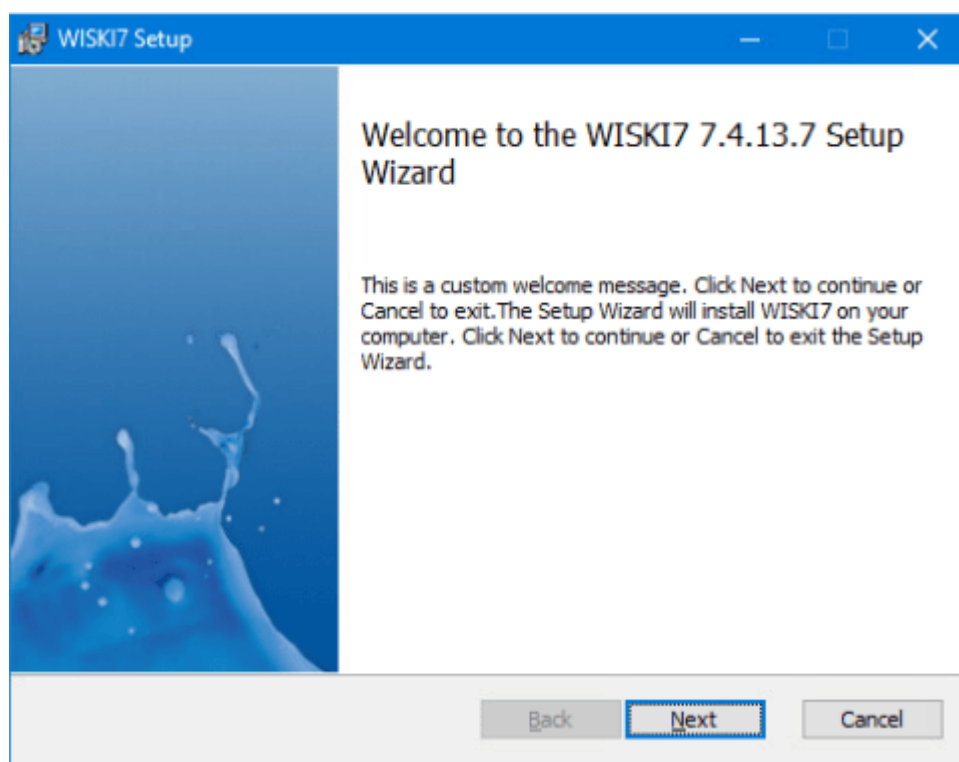
- [Installation](#) ^[209]
- [Deinstallation](#) ^[217]
- [Upgrade](#) ^[219]

19.3.1 Installation

When executing the following MSI files, you can run a client installation with the help of the WiX Toolset wizard:

- WISKI7
- KiDSM Client (up to version 7.4.11 SR 5)
- KiScriptIDE
- Service Monitor

Note: During the *Service Monitor* installation, the *JProcConsole* is installed, too.



Note:

- **From WISKI version 7.4.9 SR10:**
 - Microsoft .NET Framework version 4.5 or higher is required for WISKI7. Install a suitable version of the .NET Framework first. Otherwise the WISKI7 installation process will be aborted.
- **Up to WISKI version 7.4.11 SR13:**
 - Updating an existing client installation via the GUI can only be done by running MSI files on existing installations with the **REPAIR** option. If you wish to perform a full update, please use [silent mode](#) ^[220].
 - From version 7.4.11 SR14 to 7.4.13 SR6, you can update the WISKI7 client using the installation programme (MSI) with the help of a wizard. The old client is removed and a new version is installed.
- **Up to WISKI version 7.4.13 SR6:**
 - You can use the [HELP] button to display installation instructions. In this case, the Adobe Reader application must be installed on the computer or an error message will be displayed.

As you navigate through the wizard, one or more steps from the following chapters may be required for installation.

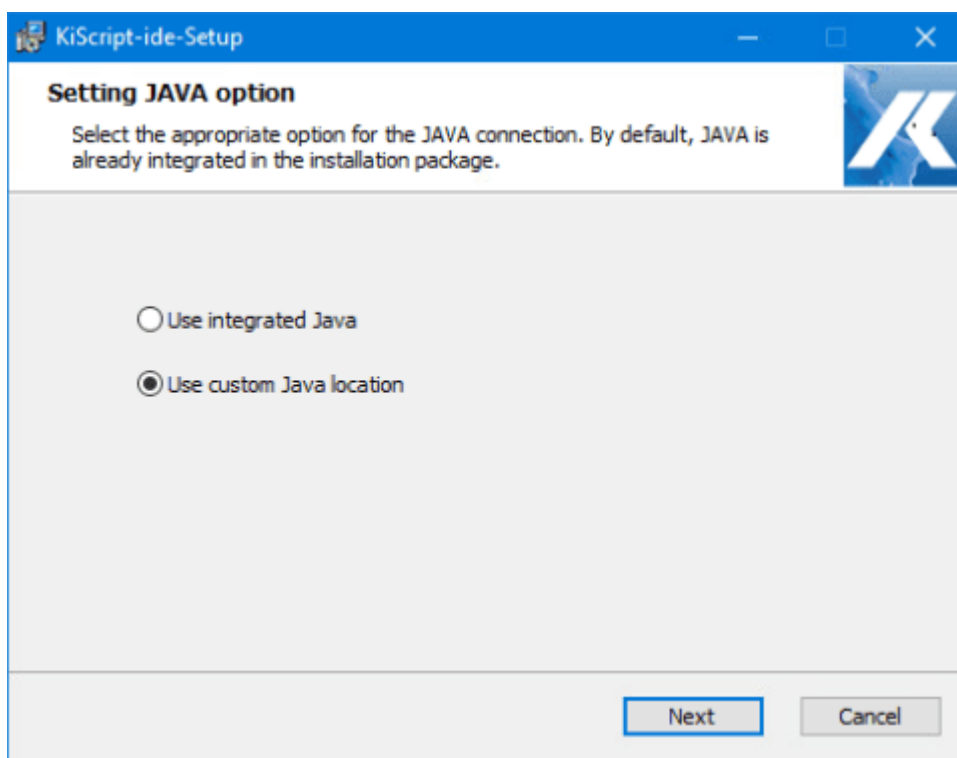
- [Java.exe Location](#) ^[210]
- [Target Installation Destination](#) ^[212]
- [Application Configuration \(WISKI7 only\)](#) ^[213]
- [Redirecting Application Configuration File \(WISKI7 only\)](#) ^[215]

- Determine Map File Directory^[216]
- Finalising^[217]

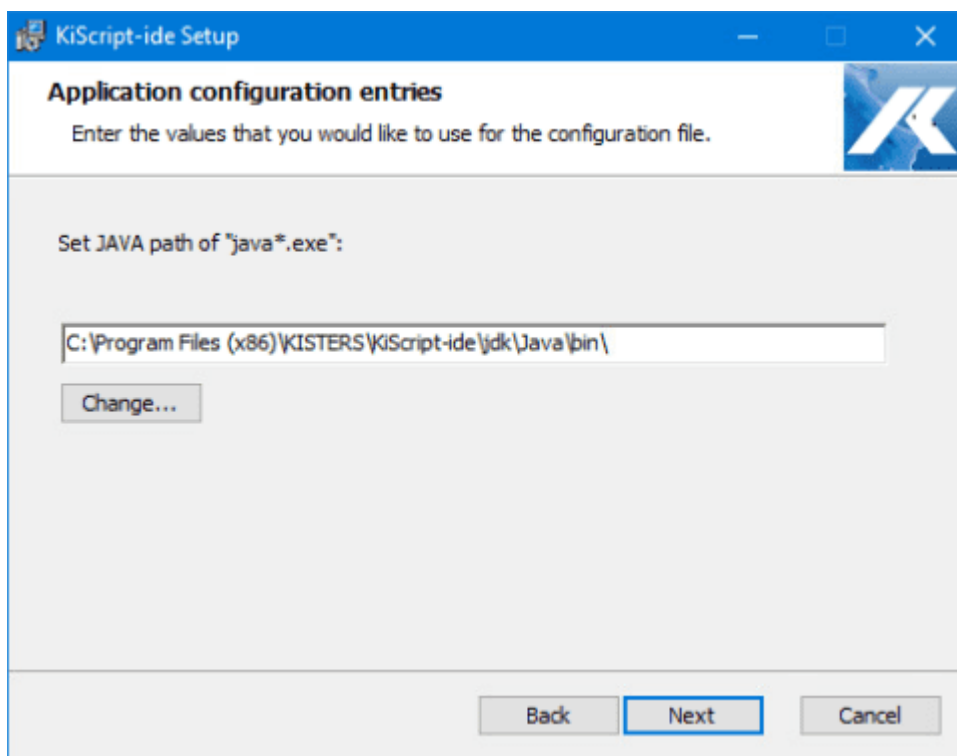
19.3.1.1 Java.exe Location

KiScriptserver-ide

- It is possible to use a custom Java location. To do so, select the list item and click **[NEXT]**.

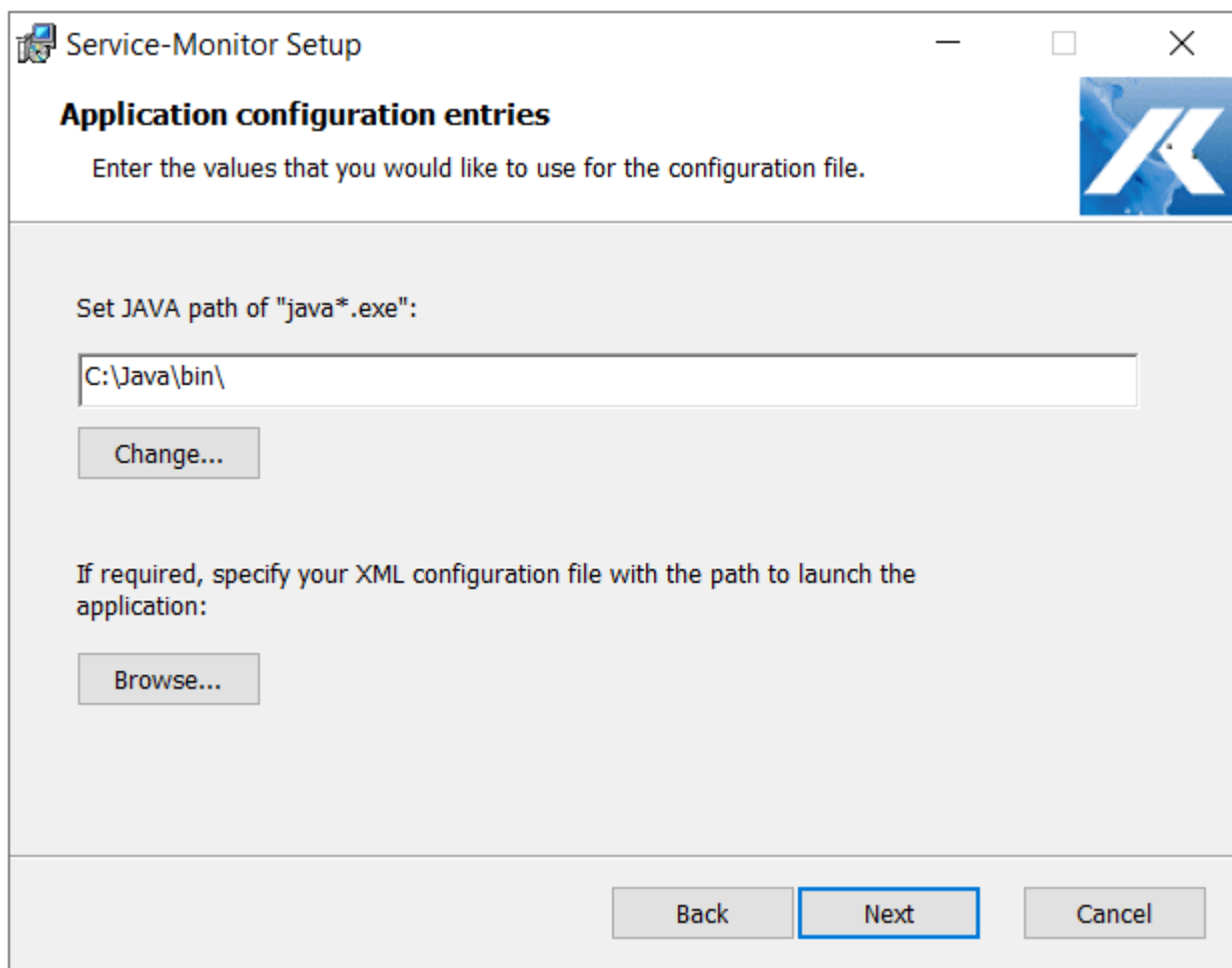


- Adjust the custom location of `java.exe` using the **[CHANGE...]** button.



Service-Monitor

- It is possible to use a custom Java location.



- Adjust the custom location of java.exe using the [CHANGE...] button.
- If you want to use a custom configuration XML file, you can do so by clicking the [BROWSE...] button.

Note:

- **Up to version 7.4.13 SR6:**
 - If the location of `java.exe` is not correct, a warning message is displayed.
 - If `JAVA_HOME` has not been configured as OS environment variable, a warning is displayed and the installation is terminated.
 - If the two policy files `local_policy.jar` and/or `US_export_policy.jar` are not found in a sub directory, an error message is displayed and the installation is terminated.
 - The check of `JAVA_HOME` and policy files is not performed from version 7.4.11 SR9. Java 11 is required on the target machine.

If set, the following configuration file contents are written according to the application.

ServiceMontior.properties

```
vm=<Path to the Java bin directory>
```

KiScriptServerIDE.ini

```
-vm
<Path to the Java bin directory>
```

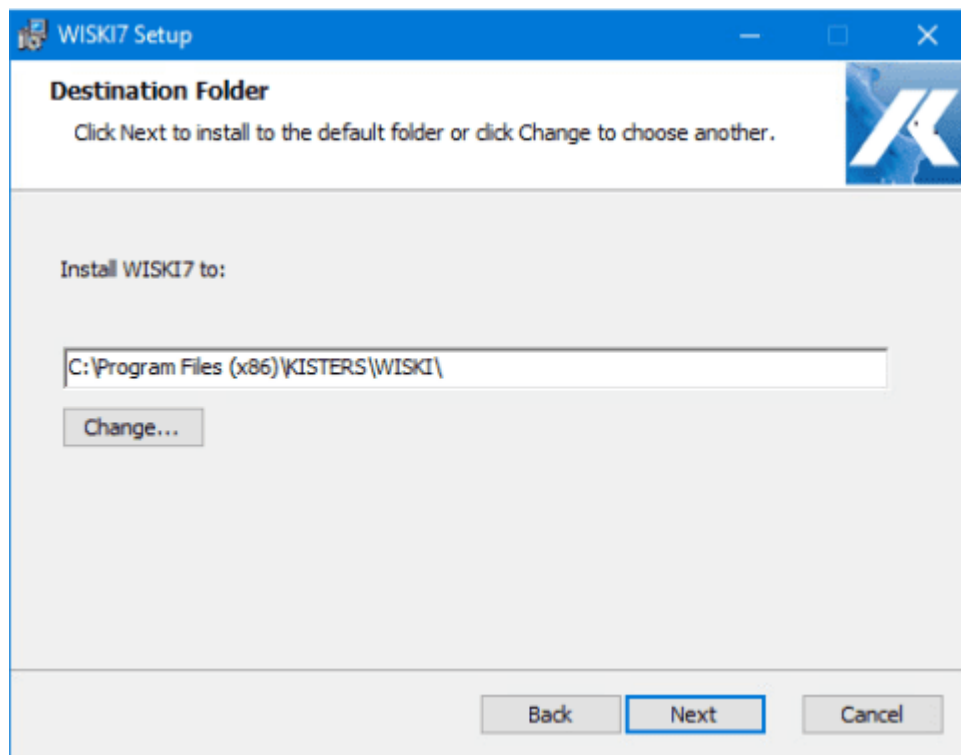
KiDSMClient.ini

```
-vm
```

```
<Path to the Java bin directory>  
-data  
@user.home/.kisters/KiDSM
```

19.3.1.2 Target Installation Destination

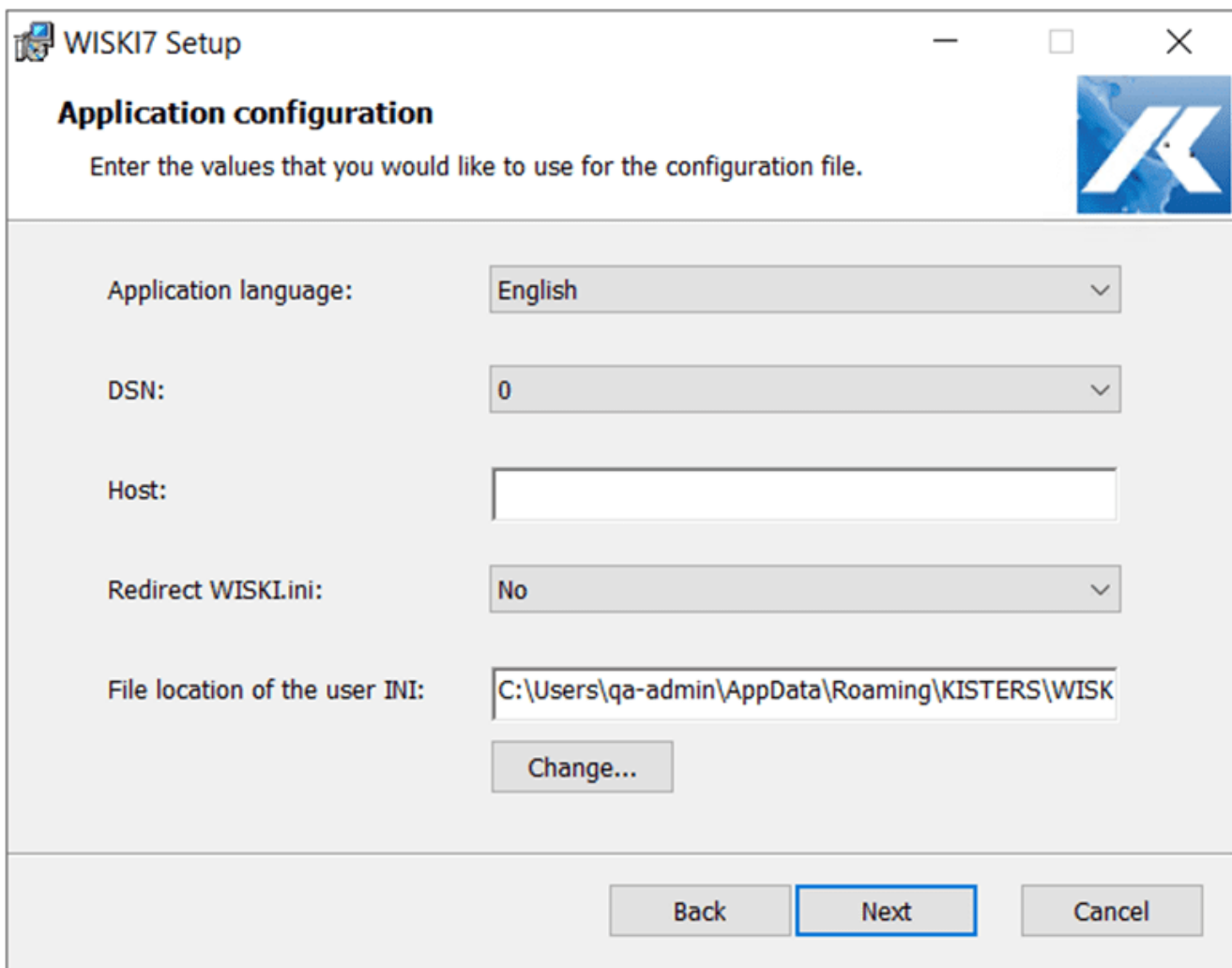
- Select a target directory for your client installation.



Note: The default destination is %ProgramFiles%. To avoid administrative problems when working with multiple non-administrator users, it is recommended that you select a different root folder, e.g. **c:\kisters**.

19.3.1.3 Application Configuration (WISKI7 only)

- Set the installation configuration for the following options.



The image shows the 'WISKI7 Setup' window with the 'Application configuration' tab selected. The window title is 'WISKI7 Setup' and it has standard Windows window controls. The main heading is 'Application configuration' with a sub-instruction: 'Enter the values that you would like to use for the configuration file.' There is a KISTERS logo in the top right corner. The configuration options are as follows:

- Application language:** A dropdown menu currently set to 'English'.
- DSN:** A dropdown menu currently set to '0'.
- Host:** An empty text input field.
- Redirect WISKI.ini:** A dropdown menu currently set to 'No'.
- File location of the user INI:** A text input field containing 'C:\Users\qa-admin\AppData\Roaming\KISTERS\WISK'. Below this field is a 'Change...' button.

At the bottom of the window are three buttons: 'Back', 'Next' (which is highlighted with a blue border), and 'Cancel'.

Application language

Here you can decide whether your WISKI7 client application should be displayed with a GUI language other than English. The language setting is written to the application configuration file (e.g. **WISKI.ini**) in the **[Translation]** section.

Default: English

Currently supported languages are: Chinese, English, German, French, Italian, Spanish

Example

```
[translation]
language=German
```

DSN

Here you can set the DSN entry for ODBC driver handling. When changed, the **DSN** entry is written to the application configuration file (**WISKI.ini**) in the **[ODBCDriver]** section.

Default: 0

Example

```
[ODBCDriver]
```

```
DSN=1
```

Host

Here you can predefine your server host. The host information is written to the user configuration file (`user_~.ini`) in the `[tsm]` section.

Default: localhost (until WISKI version 7.4.13 SR10)

Redirect application configuration file

If you want to redirect the application configuration file (`WISKI.ini`) select **YES** in the list box. See [Redirecting Application Configuration File](#)^[215] for details.

File location to user configuration file

If you adapt the path, the user configuration file is the path written to the application configuration file (`WISKI.ini`) in the `[paths]` section.

Default is

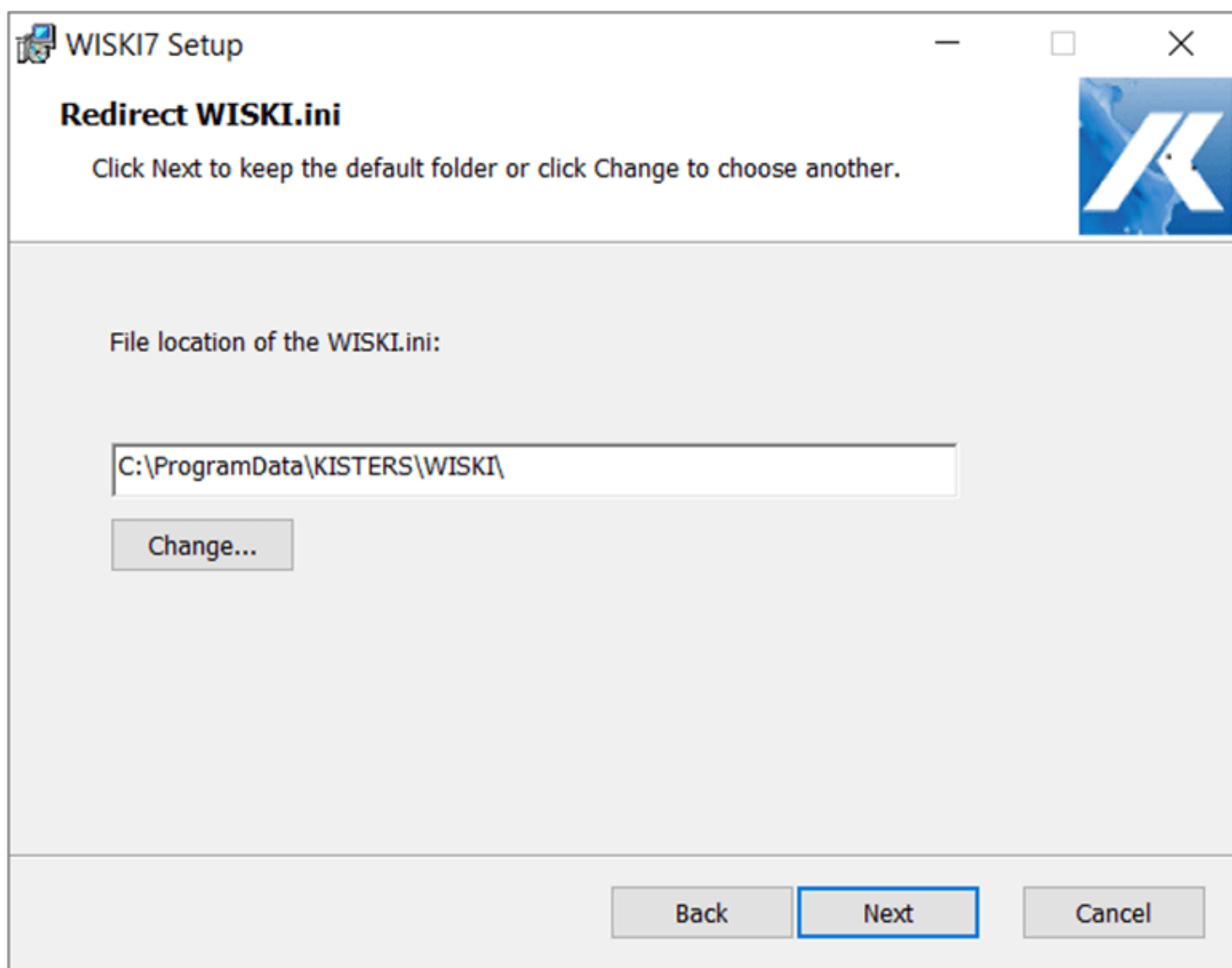
- `%AppData%\KISTERS\WISKI\` (If WISKI7 is installed in `%ProgramFiles%`)
- WISKI7 installation directory (If WISKI7 is **not** installed in `%ProgramFiles%`)

Example

```
[paths]
userinipath = C:\KISTERS\WISKI\documents\conf\wiski-client
```

19.3.1.4 Redirecting Application Configuration File (WISKI7 only)

- To use centralised server environments (terminal server farms) you can redirect the application configuration file (WISKI.ini) to a central location.

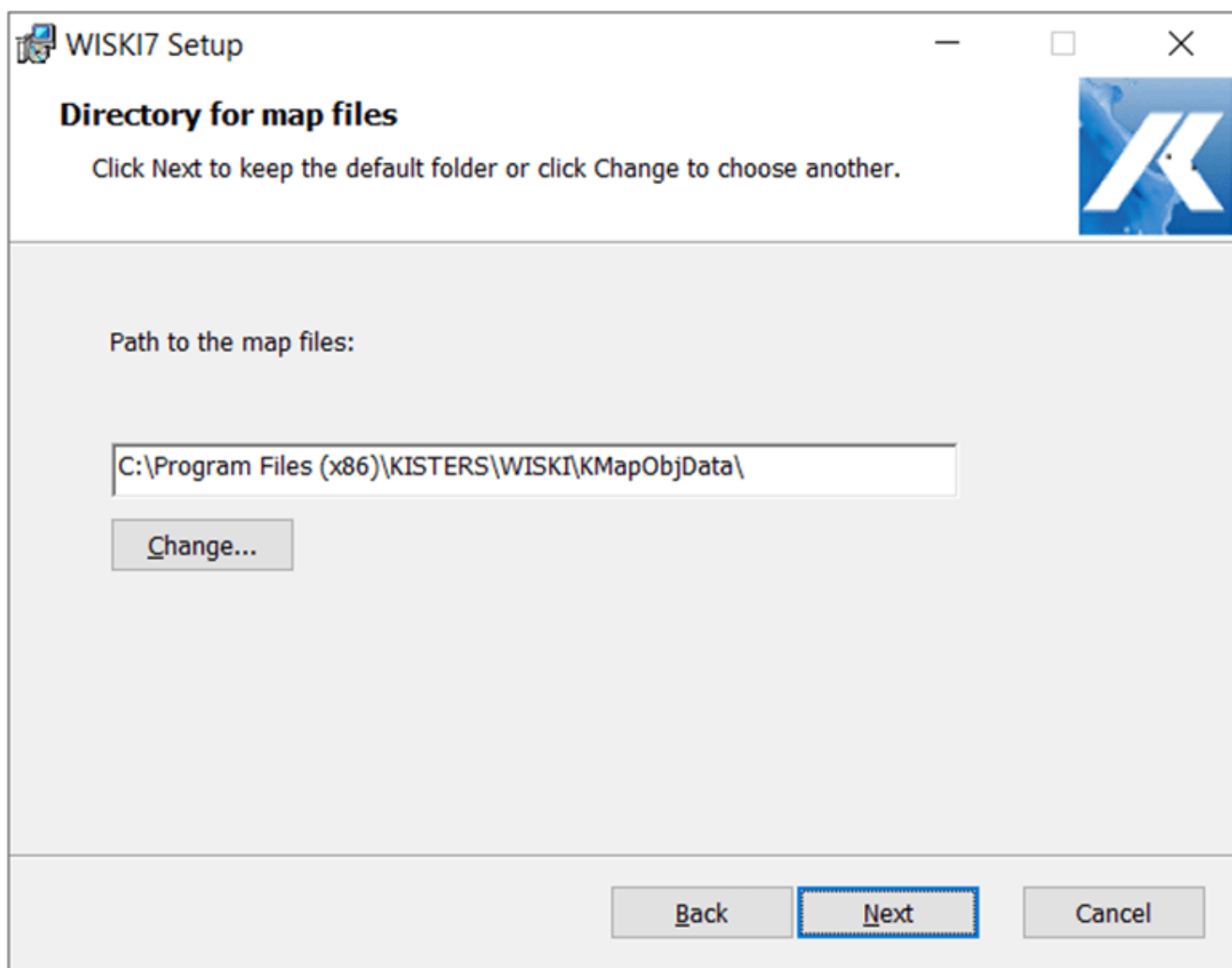


If active, a central location entry **app_ini_path** can be defined and is written to the (additionally created) **Redirect.ini** configuration file (located within the installation directory) in the **[redirect]** section.

Default: No

19.3.1.5 Determine Map File Directory

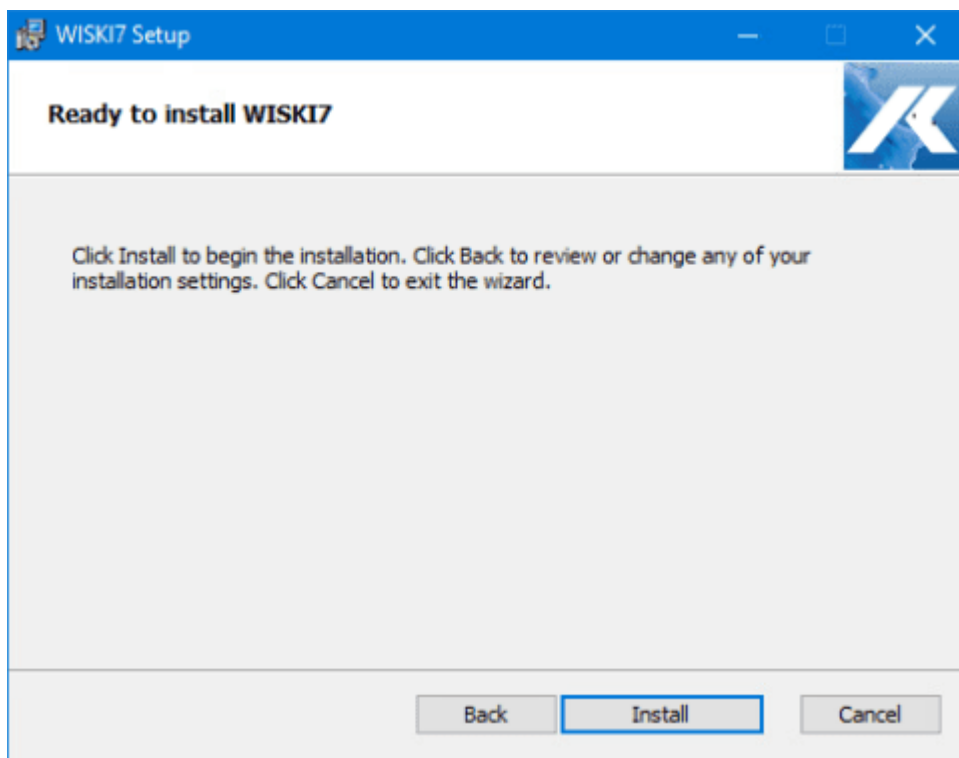
- Specify the path where the map file is located.



- To determine an individual path, click the **[CHANGE]** button and select the path in the default WINDOWS dialog for selecting files

19.3.1.6 Finalising

- After configuring the installation setup, the installation can be started using the **[INSTALL]** button.



During the installation process, the configuration setup options are written into the according configuration files (described above).

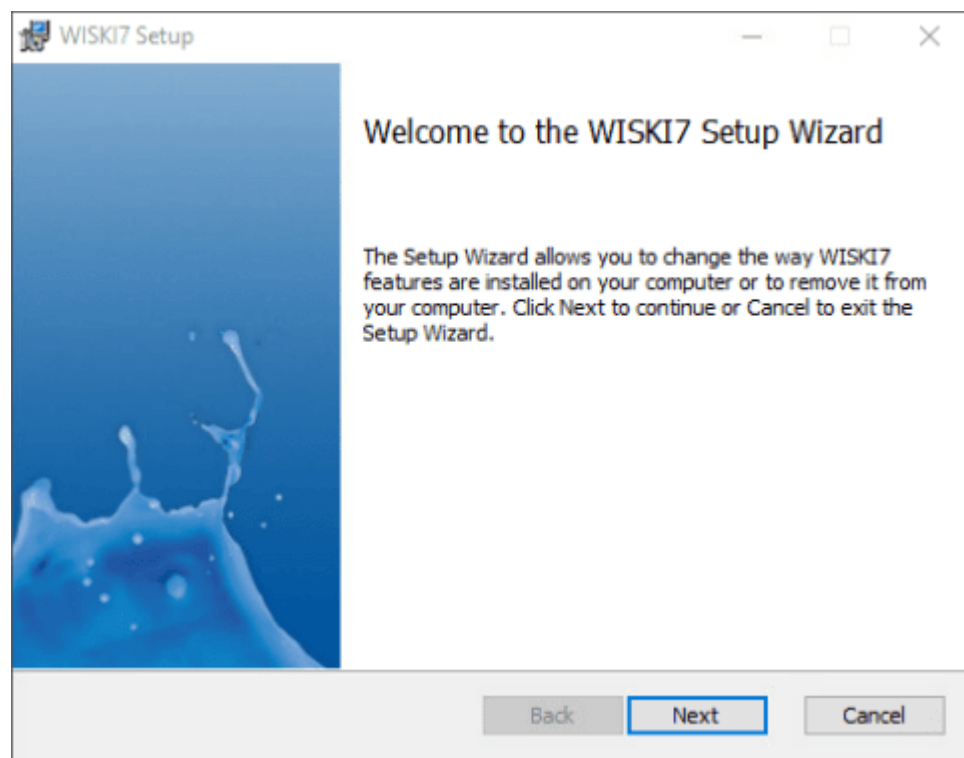
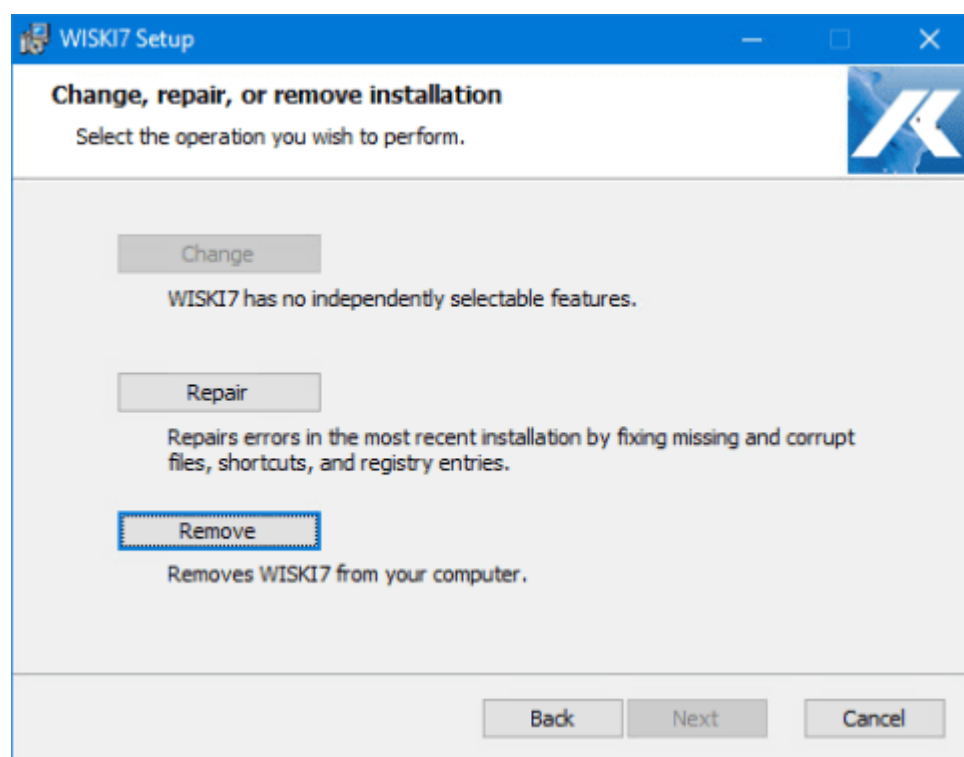
Default location of the WISKI.ini file:

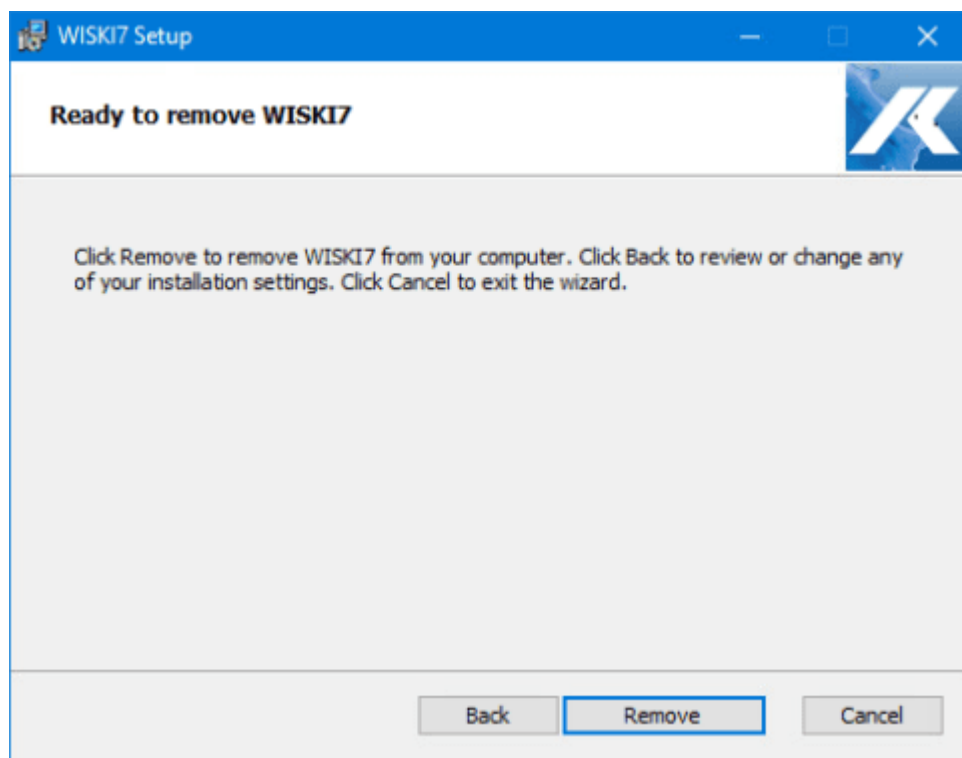
- **%ProgramData%\KISTERS\WISKI** (If WISKI7 is installed in **%ProgramFiles%**)
- WISKI7 installation directory (If WISKI7 is NOT installed in **%ProgramFiles%**)

19.3.2 Deinstallation

To remove the client application from your system, you can reuse the MSI file to launch the Uninstall Wizard (**REMOVE** option) or simply use the OS Programs and Features list by selecting the application and pressing the **[UNINSTALL]** button.

Note: Before uninstalling the application, you may need to save your custom configuration files.



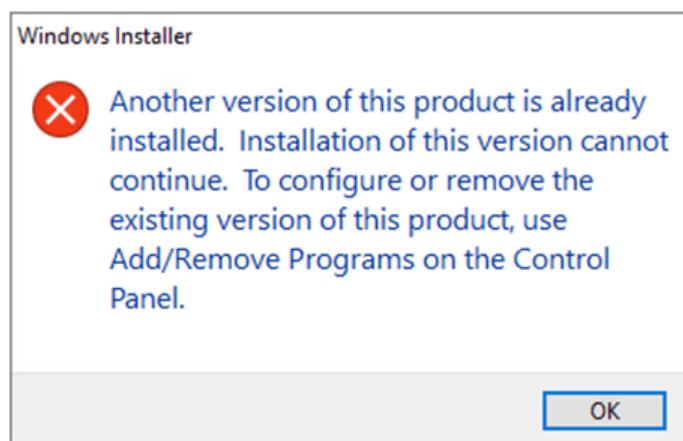


19.3.3 Upgrade

To upgrade your existing WISKI7 installation, run the new MSI file and follow the steps in the installation/upgrade wizard.

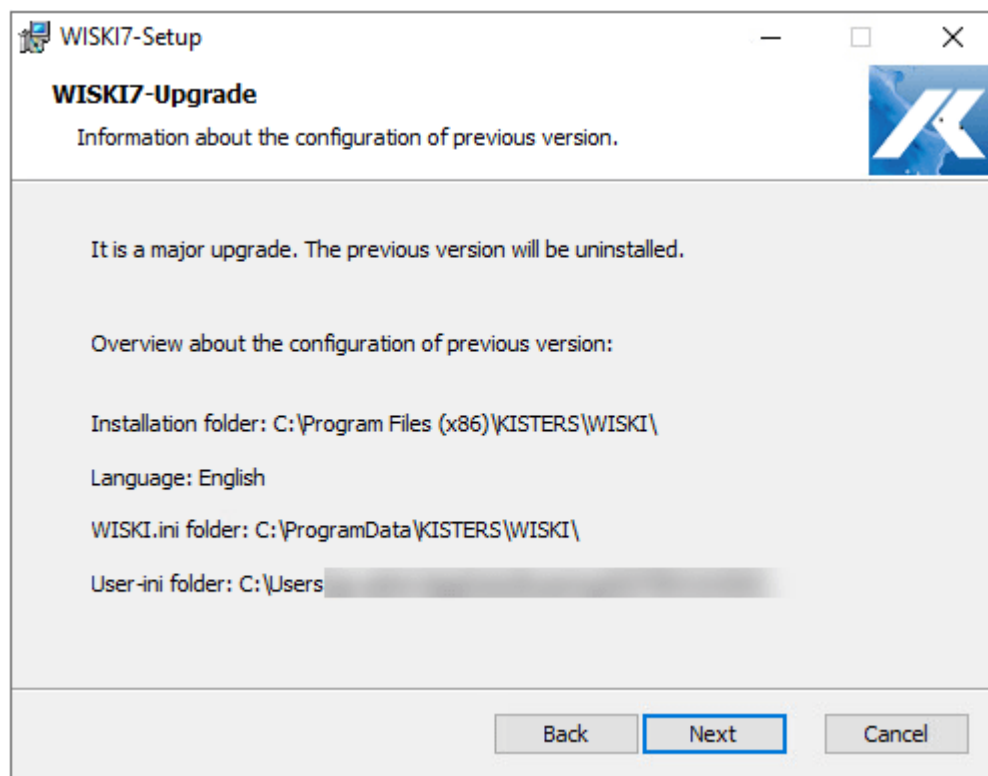
Note:

- **From WISKI version 7.4.13 SR7:**
 - When you upgrade to a new service release version, the previous installation is automatically uninstalled and the new version is installed.
 - When upgrading with a service pack or service patch to the same service release version, the previous installation should be manually uninstalled first. Otherwise, an error message will be displayed, see below. The user configuration is retained after uninstallation.



- **Up to WISKI version 7.4.13 SR7:**
 - When upgrading, you will need to uninstall the previous installation before installing the new version.

During the upgrade process, a dialogue box will be displayed with upgrade information and an overview of the current setup.



19.4 Client Installation/Update in Silent Mode

Client installation in silent mode comprises these areas of functionality:

- [Installation](#) ²²⁰
- [Update](#) ²²¹
- [Deinstallation](#) ²²³

19.4.1 Installation

To start an unattended client installation/upgrade, you will need the MSI file. Open the command prompt with administrative rights. Navigate to the appropriate directory where the MSI file is stored.

To install the program using the silent mode, enter this in the command prompt and execute:

```
>msiexec /i <client>.msi /qn [/l*v "<installation/update log file>"]  
<arguments>
```

The following arguments are available:

WISKI7 (from version 7.4.13 SR7)

Argument	Options	Description
INSTALLFOLDER	=<path>	Installation directory, default is "%ProgramFiles%\KISTERS\WISKI", e.g. INSTALLFOLDER="%ProgramFiles%". Note: If there are blanks in the path, you have to use " " .
SETUPLANGLIST	English/German/French/Spanish/Chinese/Italian	Client language, default is English .

Argument	Options	Description
SETUPDSNLIS T	=0 or =1	DSN entry in WISKI.ini (1 = yes / 0 = no, default 0). Specifies if DSN is used or not, mainly needed when using MSSQL. Note: If you do not specify this argument, no DSN entry is written in the [ODBCDriver] section in the WISKI.ini , otherwise DSN=1 is written.
SETUPHOST	=<your hostname>	Hostname of the server where WISKI7 is installed. If you do not specify this argument, no entry for the host is written.
INIUSERLOC	=<path>	If not specified, default values are used. Default values: <ul style="list-style-type: none"> Up to WISKI version 7.4.13 SR6: <ul style="list-style-type: none"> %AppData%\KISTERS\WISKI (If WISKI7 is installed in %ProgramFiles%) Installation path (If WISKI7 is NOT installed in %ProgramFiles%) From WISKI version 7.4.13 SR7: <ul style="list-style-type: none"> %AppData%\KISTERS\WISKI Note: If there are blanks in the path, you have to use " ".
SETUPREDIRAP PPFILE	=0 or =1	Default is 0. <ul style="list-style-type: none"> =0: No redirection of the application configuration file. redirect.ini is not installed. =1: Redirect the application configuration file. redirect.ini is placed in the application folder.
APPINIFOLDE R	=<path>	Specifies a different path for the application configuration file if SETUPREDIRAPPFILE=1 .

Example (from version 7.4.13 SR7)

```
>msiexec /i WISKI7.msi /qn INSTALLFOLDER="%ProgramFiles%\WISKI"
SETUPLANGLIST=Geman SETUPDSNLIST=1 SETUPHOST=wiski.hostname
INIUSERLOC="C:\Kisters\INI" SETUPREDIRAPPFILE=1 APPINIFOLDER="X:
\central"
```

KiScriptIDE (up to version 7.4.13 SR6)

- INSTALLDIR:** Installation directory, default is **%ProgramFiles%\KiScriptIDE**.
- IS_BROWSE_FILEBROWSED:** <java path>\bin\java.exe (mandatory, no longer required from 7.4.11 SR9).

Example

```
msiexec /i "KiScript-ide.msi" /qn INSTALLDIR="%ProgramFiles%
\KiScriptIDE" IS_BROWSE_FILEBROWSED="C:\Java\jdk-
1.8.0.92\bin\java.exe"
```

KiScriptIDE (from version 7.4.13 SR7)

- INSTALLFOLDER:** Installation directory, default is **%ProgramFiles%\KISTERS\KiScript-ide**.
- JAVAEXELOC:** Path of the **java*.exe**, e.g. <java path>\bin, default is **INSTALLFOLDER\jdk\Java\bin**

Example

```
msiexec /i "KiScript-ide.msi" /qn INSTALLFOLDER="C:
\KISTERS\KiScriptIDE" JAVAEXELOC="C:\Java\bin"
```

Note: By default, Java is included in the installation package. It is not necessary to specify the Java path. It is also possible for customers to use their own Java, e.g. the customer provides their own Java 8 installation.

There is an additional file **ReplaceStr.bat** in the installation folder to replace the Java path in **KiScriptServerIDE.ini** automatically during installation.

Service Monitor (up to version 7.4.13 SR6)

- **INSTALLDIR:** Installation directory, default is %ProgramFiles%\Service-Monitor.
- **IS_BROWSE_FILEBROWSED:** <java path>\bin\java.exe (mandatory)

Example

```
msiexec /i "jproc-client.msi" /qn INSTALLDIR="%ProgramFiles%\Service-Monitor" IS_BROWSE_FILEBROWSED="C:\Java\jdk-1.8.0.92\bin\java.exe"
```

Service Monitor (from version 7.4.13 SR7)

- **INSTALLFOLDER:** Installation directory, default is %ProgramFiles%\KISTERS\Service-Monitor.
- **JAVAEXELOC:** Path of the java*.exe, e.g. <java path>\bin, default is C:\Java\bin
- **CUSTOMXML:** XML configuration file with full path, e.g. <xml path>*.xml, default is no argument in the command line of the application startup

Example

```
msiexec /i "Service-Monitor.msi" /qn INSTALLFOLDER="C:\KISTERS\Service-Monitor" JAVAEXELOC="C:\Java\bin" CUSTOMXML="C:\Config\ServiceMonitor.xml"
```

19.4.2 Update

To update the program using the silent mode, enter this in the command prompt and execute:

```
>msiexec /i <client>.msi /qn [/l*v "<update log file>"]
```

You have to specify the following arguments, which have to be put after /qn

WISKI7 (up to version 7.4.11 SR14)

- **REINSTALLMODE:** vamus
 - **a:** Force all files to be reinstalled, regardless of checksum or version.
 - **u:** Rewrite all required registry entries from the Registry Table that go to **HKEY_CURRENT_USER** or **HKEY_USERS**.
 - **m:** Rewrite all required registry entries from the Registry Table that go to **HKEY_LOCAL_MACHINE**.
 - **s:** Reinstall all shortcuts and re-cache all icons overwriting any existing shortcuts and icons.
 - **v:** Use to run from the source package and re-cache the local package.

Note: Do not use the **v** reinstall option code for the first installation of an application or feature.
- **REINSTALL:** ALL

Example

```
msiexec /i "WISKI7.msi" /qn REINSTALLMODE=vamus REINSTALL=ALL
```

KiScriptIDE

- **REINSTALLMODE:** vamus
 - **a:** Force all files to be reinstalled, regardless of checksum or version.
 - **u:** Rewrite all required registry entries from the Registry Table that go to **HKEY_CURRENT_USER** or **HKEY_USERS**.
 - **m:** Rewrite all required registry entries from the Registry Table that go to **HKEY_LOCAL_MACHINE**.
 - **s:** Reinstall all shortcuts and re-cache all icons overwriting any existing shortcuts and icons.
 - **v:** Use to run from the source package and re-cache the local package.

Note: Do not use the **v** reinstall option code for the first installation of an application or feature.

- **REINSTALL:** ALL
- **IS_BROWSE_FILEBROWSED:** <java path>\bin\java.exe (mandatory, no longer required from 7.4.11 SR9)

Example

```
msiexec /i "KiScript-ide.msi" /qn REINSTALLMODE=vamus REINSTALL=ALL
IS_BROWSE_FILEBROWSED="C:\Java\jdk-1.8.0.92\bin\java.exe"
```

Service Monitor

- **REINSTALLMODE:** vamus
 - **a:** Force all files to be reinstalled, regardless of checksum or version.
 - **u:** Rewrite all required registry entries from the Registry Table that go to **HKEY_CURRENT_USER** or **HKEY_USERS**.
 - **m:** Rewrite all required registry entries from the Registry Table that go to **HKEY_LOCAL_MACHINE**.
 - **s:** Reinstall all shortcuts and re-cache all icons overwriting any existing shortcuts and icons.
 - **v:** Use to run from the source package and re-cache the local package.
- Note:** Do not use the **v** reinstall option code for the first installation of an application or feature.
- **REINSTALL:** ALL
 - **IS_BROWSE_FILEBROWSED:** <java path>\bin\java.exe (mandatory)

Example

```
msiexec /i "jproc-client.msi" /qn REINSTALLMODE=vamus REINSTALL=ALL
IS_BROWSE_FILEBROWSED="C:\Java\jdk-1.8.0.92\bin\java.exe"
```

19.4.3 Deinstallation

To uninstall the program using the silent mode, enter this in the command prompt and execute:

```
>msiexec /x WISKI7.msi /qn [/l*v "<uninstallation log file>"]
```

For further command line options for **msiexec** see [this page](#).

19.5 Usage of Elasticsearch

[Elasticsearch](#) is a JSON-based search and analytics engine.

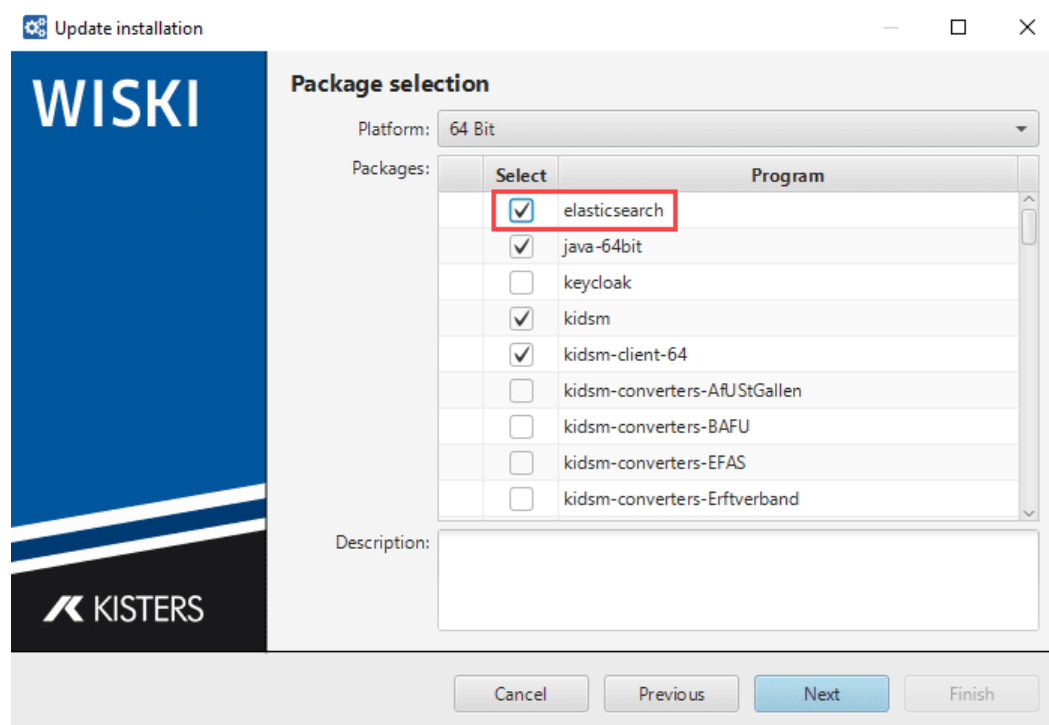
Refer to the official [elasticsearch](#) reference (<https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>) for more detailed information.

The usage of Elasticsearch comprises these areas of functionality:

- [Installation](#) ²²⁴
- [Start Elasticsearch](#) ²²⁶
- [Update Elasticsearch](#) ²²⁶
- [Find Elasticsearch in the Service Registry](#) ²²⁷

19.5.1 Installation

Every WISKI7 release comes with an optional Elasticsearch package that can be installed using the WSM.



The distribution contains a JProcController configuration file and the Elasticsearch binary folder:

- `wiski/services/jproccontroller/configuration/08-jproc-elasticsearch-default.properties`
- `wiski/services/elasticsearch/`

Basic Configuration

You configure the Elasticsearch server in the `elasticsearch.yml` file located in `elasticsearch/config/`.

For example, you can change the Elasticsearch HTTP port (default 9200) by altering the `http.port` property (see below).

`elasticsearch.yml`

```
http.port: 9200
```

Memory Settings

The configuration of the memory settings is important. You can configure them in the `jvm.options` file located in `elasticsearch/config/`.

Choose the memory settings depending on the requirements for the database. For an installation with a lot of data it might be useful to increase the memory settings. This can be done by altering the `-Xms` and `-Xmx` settings (see example below). More information about setting up the Elasticsearch memory can be found here: <https://www.elastic.co/guide/en/elasticsearch/reference/current/heap-size.html>.

Note: Always set the minimum and maximum JVM heap size to the same value.

Example:

```
jvm.options
```

```
-Xms2g
```

```
-Xmx2g
```

Linux Configuration

Additional system configuration might be necessary when running Elasticsearch on Linux.

To fix file descriptor or memory issues add the following configuration to the `/etc/sysctl.conf` file:

```
/etc/sysctl.conf
```

```
vm.max_map_count=262144
fs.file-max=65536
```

This change is permanent, but will only be active when rebooting the system. If you do not want to reboot, execute the following commands:

```
sudo sysctl -w fs.file-max=65536
sudo sysctl -w vm.max_map_count=262144
```

Note: This is not permanent. Make sure you edit the configuration file as described above.

If you want to have a user-specific configuration, you can edit the `/etc/security/limits.conf` file (see below).

```
/etc/security/limits.conf
```

kisters	soft	nofile	65536
kisters	hard	nofile	65536

Note: These settings will be active when you create a new shell.

In order to apply the changes to a system service, add the following file (`/etc/systemd/system/wiskiserver.service.d/override.conf`) to the file system and afterwards reload the daemon (`sudo systemctl daemon-reload`).

```
/etc/systemd/system/wiskiserver.service.d/override.conf
```

```
[Service]
# Specifies the maximum file descriptor number that can be opened
# by this process
LimitNOFILE=65536

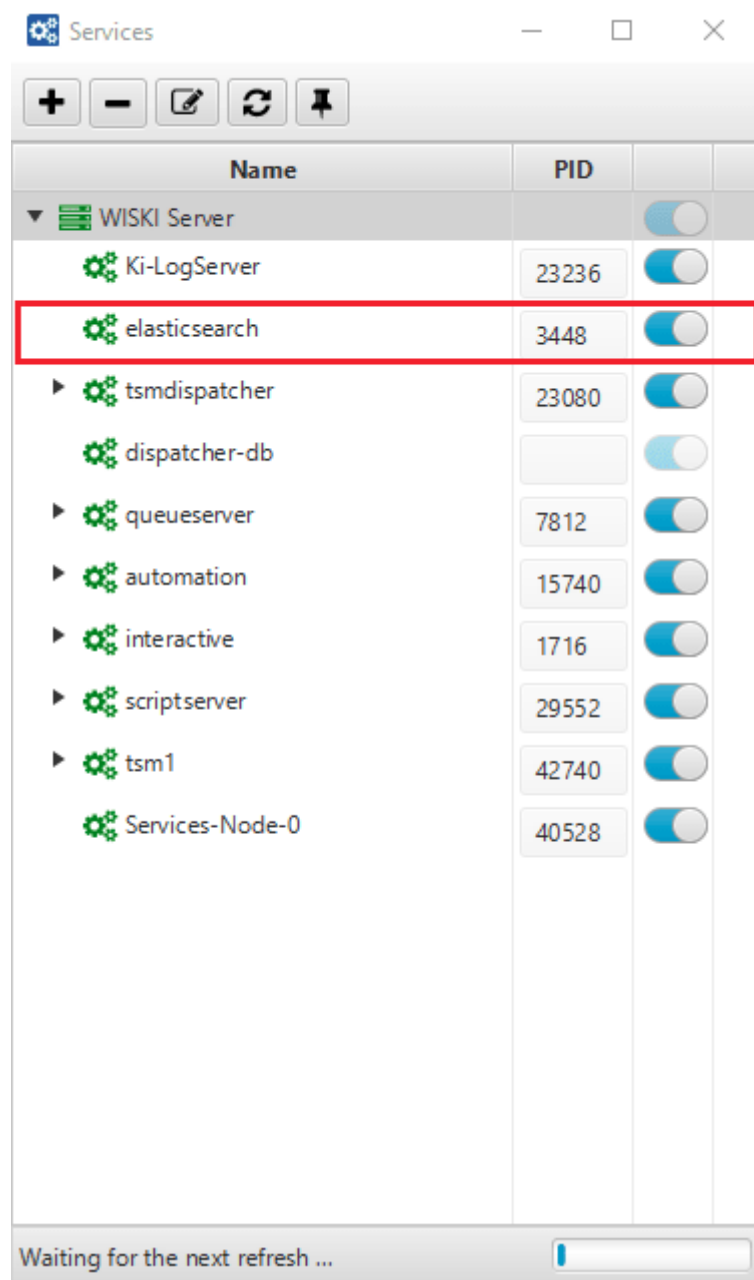
# Specifies the maximum number of processes
LimitNPROC=4096

# Specifies the maximum size of virtual memory
LimitAS=infinity

# Specifies the maximum file size
LimitFSIZE=infinity
```

19.5.2 Start Elasticsearch

Once Elasticsearch is installed, it automatically starts and stops with the WISKI7 server. The status of Elasticsearch can be viewed within the services view of the WSM (see screenshot below).



19.5.3 Update Elasticsearch

Note that there is a major change in Elasticsearch between versions 5 and 7, which requires additional steps explained below. This affects WISKI distributions prior to 7.4.13 SR7, as these versions contain Elasticsearch 5, while 7.4.13 SR7 introduces Elasticsearch 7.

If Elasticsearch indexes need to be retained (e.g. for the Process Analytics and Application Metrics web applications), follow the steps below. Do this before updating WISKI.

- Make sure you are running Elasticsearch version 5.
- Run

```
java -jar PortalServerEsMigrationExec.jar mode=EXPORT
baseUrl=http://localhost:9200/
```

```
basePath=C:/es_migration/2022/export
```

Replace `C:/es_migration/2022/export` with an appropriate directory name.

- Update WISKI and Elasticsearch to version 7.
- Make sure that the Water Portal and Process Analytics importers are not yet started.
- Run

```
java -jar PortalServerEsMigrationExec.jar mode=IMPORT
baseUrl=http://localhost:9200/
basePath=C:/es_migration/2022/export
```

If you do not need to retain Elasticsearch indexes, follow the steps below:

- Stop Elasticsearch using the Service Monitor.
- Make a backup of the Elasticsearch folder. This is usually the path: `\KISTERS\documents\elasticsearch\data`.
- Start Elasticsearch using the Service Monitor.

19.5.4 Find Elasticsearch in the Service Registry

Elasticsearch is registered in the Service Registry. You can query the WISKI REST API for the Elasticsearch port or host using the service ID of Elasticsearch.

Elasticsearch ID in Service Registry:	DE.KISTERS.WATER.DATABASE.ELASTICSEARCH
<code>http://localhost:7420/serviceregistry/v2/apps/DE.KISTERS.WATER.DATABASE.ELASTICSEARCH</code> or	
<code>https://localhost:7421/serviceregistry/v2/apps/DE.KISTERS.WATER.DATABASE.ELASTICSEARCH</code>	
REST API query:	
<code>http://localhost:7480/rest/water/registry/services/de.kisters.water.database.elasticsearch/single</code> or	
<code>https://localhost:7479/rest/water/registry/services/de.kisters.water.database.elasticsearch/single</code>	

19.6 Python as Scripting Language

Core Functionality

The Python Framework for Reporting and Operations is based on a custom portable Miniconda distribution and offers you a variety of advantages compared to older scripting languages.

- A general purpose, open source language, which is widely spread not only in the IT sector but especially in science and engineering.
- A variety of ready for use libraries for data analysis, web frameworks, statistics, modelling and AI.
- An active community for support requests.
- Beginner-friendly – easy to read and adapt the code.
- Integration in web applications.
- Top programming language ahead of JavaScript and C++.

Features

The Python Framework for Reporting and Operations can currently be used for:

- **Data retrieval** (time series, water quality data and any meta data) via the server (REST-API and KiWIS) for any post-processing outside of WISKI.

- **Reports** that can be integrated and accessed with the Report Wizard.
- **Operations:** With Python, individual operations can be created and integrated into the WISKI system view.
- **Data validation:** To extend the standard WISKI system, an external framework based entirely on Python is available for user-specific and complex validation routines.

Installation and Configuration

To install the Python Framework for Reporting and Operations, you must select certain packages in the WISKI Server Manager during the installation or update process. See [Package selection](#)^[65].

19.7 FAQ

A general hint at the beginning: Check also the FAQ appendix of the official WISKI user documentation (online help or PDF) included in your installation or talk to your system administrator if connection difficulties, installation or other problems occur.

The FAQ section covers the following topics:

- [FAQ on TLS and Certificate Management](#)^[228]
- [FAQ on Keycloak](#)^[232]

19.7.1 FAQ on TLS and Certificate Management

A general hint at the beginning: Check also the FAQ appendix of the official WISKI user documentation (online help or PDF) included in your installation or talk to your system administrator if connection difficulties, installation or other problems occur.

When did we introduce TLS / Certificate support in WISKI?

TLS/certificate support was initially introduced with 7.4.11 SR7.

Which versions of TLS does WISKI support?

See the first chapter of this document.

[WISKI Server](#)^[125] supports and requires at least TLS 1.2.

TLS 1.3 is supported too but currently untested (= not tested by our QA).

Any older version will be refused by WISKI Server. See also the `jdk.tls.disabledAlgorithms` key in `wiski/services/java/conf/security/java.security`.

This should apply for all of our Java backend modules.

[WISKI Desktop](#)^[127] supports TLS 1.2.

TLS 1.3 is completely untested and currently not supported for REST calls via PowerBuilder.

Which version is allowed or not is defined on the operating system layer.

What do I have to provide as a customer?

Be aware of the issues listed hereafter, if you cannot or do not want to provide your own server certificate signed by an official and/or trusted Certification Authority (CA):

- Accessing WISKI Server HTTPS resources (like the Standard Portal or others) with a browser will require additional effort like importing the generated self-signed root CA into the operating system's TrustStore on all client machines. Not doing so will lead to issues.
- Handling of so called "[WISKI Server Satellites](#)^[115]" will be more complicated as it must be ensured that the whole system uses one single generated self-signed root CA.
- Your system is not as secure as it could and should be, since our generated self-signed root CA could be misused to sign other server certificates being then trusted on all other WISKI client machines.

We strongly recommend to use a server certificate signed by an official and/or trusted Certification Authority!

If you still prefer to use the generated self-signed root CA provided by default, you can stop reading here - which we do not recommend!

If you want your own certificates to be added to our WISKI Server KeyStore and TrustStore, provide the following files to us using the installation/update wizard dialog of the WISKI Server Manager:

- X.509 Root Certification Authority Certificate
File format: PEM
- X.509 Server Certificate - including the whole certificate chain up to the root CA
File format: PEM
If and only if the server certificate has a X.509 Extended Key Usage extension: the purposes TLS Web Server Authentication and TLS Web Client Authentication (for RMI) must both be set!
- Server Certificate Private Key
File format: PEM
Private key format: PKCS #8 or PKCS #1
Private key algorithm: EC or RSA

Alternatively, you can also provide the KeyStore and TrustStore files by yourself. The stores must be in PKCS #12 format and must be password-protected.

What is a self-signed certificate?

A self-signed server certificate is typically signed by itself (aka the developer or company, anyone) or - as implemented in WISKI Server - by an untrusted root certification authority which is generated by the WISKI Server setup (Common Name **Generated Self-Signed Root CA for WISKI Server <host>**).

Some customers as well as KISTERS SysAdmins make also usage of self-signed certificates for internal usage. This is cheaper and more flexible than using certificates signed by an official certification authority. Typically the generated root certification authority used to sign such server certificates is then being rolled out automatically to all machines' operating system TrustStore in the network. This is not the case when using self-signed certificates signed by a generated certification authority as used by default by the WISKI Server.

An official server certificate is being signed by an official root certification authority, e.g. GlobalSign, which is globally trusted by all operating systems TrustStores and which typically takes money for that.

What's a certificate chain?

A certificate chain is an ordered list of certificates stored in one file starting with the server certificate via it's (optional) intermediate certificates to the root CA certificate.

Such a chain can be build manually by concatenating e.g. server.crt and rootCA.crt in the defined order or automatically by using the [Security Tool](#)^[154].

Most server processes consume a server certificate chain: directly by consuming a chain in e.g. a PEM file like chain.crt or by consuming a KeyStore which contains in addition to the chain also the protected server private key. The last mechanism is being used by WISKI Server.

The server certificate chain will be presented to each client connecting to a TLS secured port. The server certificate will be used by the client to verify, among other, that the DNS name matches - you need to be sure to talk with the correct secured endpoint and not something else when presenting e.g. a password, the certificate is valid for the required use case and not expired. Intermediate till root CA certificates will be used by the client to ensure the remote endpoint is trusted by validating the presented root CA with it's counterpart in the client TrustStore.

My system has multiple hostnames. How to handle that?

Certificates store the mandatory system FQDN DNS name into the **CommonName** field. Additional FQDN names can be specified using the Certificate Subject Alternative Name Extension.

The [WISKI Server Security Tool](#)^[154] is able to generate self-signed certificates using multiple hostnames.

Example:

```
cd services/wiski-server/bin/
../../../../java/bin/java -jar security-tool.jar createCSR --commonName
hostname.example.kisters.de --altDNSNames alternative-
hostname.example.kisters.de,alternative-hostname2.example.kisters.de
../../../../java/bin/java -jar security-tool.jar signCSR
../../../../java/bin/java -jar security-tool.jar buildCertificateChain
```

The server private key can be found in **services/config/local/unmanaged/security/server.key**.

The server certificate chain can be found in **services/config/local/unmanaged/security/chain.crt**.

Both can now be added to the KeyStore or be used for another system using the same root CA.

This is intended to be used for machines with multiple hostnames.

In theory, you could also use the same mechanism to provide one certificate for multiple machines, but this is not recommended as you would also share the same private key for all machines. If one is compromised, all of them are.

My certificate expired. What should I do?

Each certificate (server certificate, intermediate CA certificate or root CA certificate) has an expiration time. The lifetime of certificates is becoming shorter and shorter over time as security requirements get stronger and stronger.

Currently, the best way to check the validity of the server certificate chain is to use the system browser by calling e.g. **https://localhost:7479**.

replace **localhost** by your FQDN hostname to avoid TLS warnings in the browser.

The certificates can then be inspected by using buttons on the left of the browser URL, the exact implementation differs and depends on the browser being used.

Finally, the whole server certificate chain will be displayed by the browser and each single extension can be examined, for example for the expiration time.

See also <https://www.globalsign.com/en/blog/how-to-view-ssl-certificate-details> for more precise instructions about how to do that for different browsers.

There is currently no automated check within WISKI Server, no automated renewal for the default self-signed certificates and no way to read the expiration time with our tooling. All of that is however on our list.

If the server certificate expires or is coming close to expiration time, it must be renewed. In case our default self-signed certificates are being used, just regenerate the certificate using our [Security Tool](#):

- Generate a new Certificate Sign Request (CSR) - check the Security Tool documentation for customisation:

```
../../../../java/bin/java -jar security-tool.jar createCSR
```

The command will also regenerate a new server keypair (private and public keys).

Note:

- Typically the keypair is not regenerated for a simple renewal but this is currently not supported by our tooling. Also, keypair regeneration should imply the revocation of the previous certificate, but this is also currently not supported.
- In case a satellite node is affected, the **createCSR** command must also contain the satellite hostname as argument for the CommonName field of your certificate which represents the FQDN host name hostname host name :

```
../../../../java/bin/java -jar security-tool.jar createCSR --
commonName <SATELLITE_HOSTNAME>
```

- Sign the new CSR with our self-signed root CA - again, check the documentation for customisation:

```
../../../../java/bin/java -jar security-tool.jar signCSR
```

Note: In case a [satellite](#)^[115] node is affected, the **signCSR** command must be executed on the master node because the self-signed root CA private key is stored there. The resulting **server.crt** will be stored in **services/config/local/unmanaged/security/**.

- Generate a new server certificate chain:

```
../../../../java/bin/java -jar security-tool.jar buildCertificateChain
```

Note: In case a [satellite](#)^[115] node is affected, this command must be executed on the satellite node itself. The **server.crt** generated in the previous step must be **copied from the master node to the satellite node**.

- Finally, import the new server certificate to the KeyStore with the same alias than the previous one:

```
../../../../java/bin/java -jar security-tool.jar addCertToKS
```

- Restart WISKI Server.

If the self-signed root CA certificate is about to expire, follow the steps below to renew it:

- Generate a new root CA certificate - on the master node only:

```
../../../../java/bin/java -jar security-tool.jar genRootCA
```

- As all server certificates signed by the previous expired self-signed root CA are now invalidated too, you need to regenerate new server certificates too. Follow the instructions above.

Note: In case a multi-node system is being used, you must copy the new **rootCA.crt** to all satellite nodes!

Note: If you, as explained above, configure and use your own certificates, make sure, together with your IT administration, to provide up-to-date certificates! They can be updated in WISKI Server the same way than they are configured the very first time: Use the WISKI Server Manager or use our [Security Tool](#)^[154]. There, check especially the commands to build a certificate chain and to add required certificates to the WISKI Server KeyStore and TrustStore.

How to check if the certificate is trusted in Windows?

WISKI Server provides a small utility to test HTTPS connections. As the tool is a native non-Java tool, it uses the Windows TrustStore by default.

The tool is available at **services/wiski-server/bin/httpstest.exe**.

It allows to specify a URL to test and optionally the TLS protocol version to be used.

Examples:

Successful test of HTTPS connection against TSM Dispatcher Tomcat with correctly configured Windows TrustStore:

```

C:\Windows\System32\cmd.exe

C:\kisters\wiski\services\wiski-server\bin>httpstest.exe
Syntax: httpstest <URL> [TLS version]
TLS version may be AUTO, 1.1, 1.2 (default) or 1.3

C:\kisters\wiski\services\wiski-server\bin>httpstest.exe https://vm-wprj-wupper.kisters.de:7421/
Since not configured, the default TLS protocol version 1.2 will be used
Request OK:
<!-- Copyright (c) KISTERS, Aachen 2008 -->

<head>
  <meta http-equiv="expires" content="0">
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <link rel="SHORTCUT ICON" href="kisters-favicon.ico">
  <link rel="stylesheet" title="Default" href="k4.css" media="all">
  <title>KISTERS</title>

  <meta http-equiv="refresh" content="0;URL=TsmSAdmin">
</head>
<body>
<table width="100%" height="100%" cellpadding="0" cellspacing="0">
<tr><td width="3%" height="85" bgcolor="#0060A7"></td><td height="85" bgcolor="#0060A7"><a href="http://www.kisters.de">
</a></td></tr>
<tr><td colspan="2" width="100%" height="15" bgcolor="#000000"></td></tr>
<tr><td bgcolor="#DCDDDE"></td><td align="center" bgcolor="#DCDDDE" valign="top">
<br>

```

As we can see, the connection can be established and the remote content is successfully read. The conclusion is therefore, that the Windows TrustStore provides the required root CA certificates to trust the remote endpoint.

Failed test of HTTPS connection against remote TSM Dispatcher Tomcat with missing root CA certificate in Windows TrustStore:

```

C:\Windows\System32\cmd.exe

C:\kisters\wiski\services\wiski-server\bin>httpstest.exe https://vm-went-ww.kisters.de:7421/
Since not configured, the default TLS protocol version 1.2 will be used

Unhandled Exception: System.AggregateException: One or more errors occurred. ---> System.Net.Http.HttpRe
questException: An error occurred while sending the request. ---> System.Net.WebException: The underlyin
g connection was closed: Could not establish trust relationship for the SSL/TLS secure channel. ---> Sys
tem.Security.AuthenticationException: The remote certificate is invalid according to the
validation procedure.
   at System.Net.TlsStream.EndWrite(IAsyncResult asyncResult)
   at System.Net.PooledStream.EndWrite(IAsyncResult asyncResult)
   at System.Net.ConnectStream.WriteHeadersCallback(IAsyncResult ar)
   --- End of inner exception stack trace ---
   at System.Net.HttpWebRequest.EndGetResponse(IAsyncResult asyncResult)
   at System.Net.Http.HttpClientHandler.GetResponseCallback(IAsyncResult ar)
   --- End of inner exception stack trace ---
   --- End of inner exception stack trace ---
   at System.Threading.Tasks.Task.ThrowIfExceptional(Boolean includeTaskCanceledExceptions)
   at System.Threading.Tasks.Task.Wait(Int32 millisecondsTimeout, Cancellation token cancellationToken)
   at HTTPSTest.HttpsTest.Main(String[] args)

C:\kisters\wiski\services\wiski-server\bin>_

```

19.7.2 FAQ on Keycloak

A general hint at the beginning: Check also the FAQ appendix of the official WISKI user documentation (online help or PDF) included in your installation or talk to your system administrator if connection difficulties, installation or other problems occur.

How to test the LDAP user login without involving WISKI Server?

Login to the Keycloak User Console. If you are able to log in, LDAP authentication was successful.

The Keycloak Web User Console is available at <https://localhost:7415/admin/kisters-water-default/console> if the default realm (**kisters-water-default**) is being used. Each Keycloak or LDAP user is allowed to log in here. It's pretty useful to check e.g. the validity of the LDAP configuration without involving WISKI Server.

Note: It is not possible to log in with the Keycloak admin user on this page.

Can I drop and re-create the Keycloak database schema?

Prior to WISKI 7.4.13 SR11 this is possible if no customer data is lost - this must be confirmed by development team and should therefore be avoided whenever possible.

After WISKI 7.4.13 SR11 this is no longer possible as it would break the mapping with the WISKI users in the **UAD_USER** TCA DB table.

May I apply changes to the Keycloak database on my own?

To be clear, **no**.

Note: Applying changes to the Keycloak database yourself is probably the reason why one of your next WISKI updates will fail. Even if a change to the DB seems to help today, it may break things or future updates. It also means that such changes can result in data loss if backups are not available or the problem is discovered much later.

Does WISKI store my user passwords?

- If you are not using Keycloak user federation, e.g. LDAP or Active Directory, user passwords are stored as hashes in the Keycloak database.
- If your users are provided by a Keycloak user federation and therefore managed in e.g. LDAP or Active Directory, neither Keycloak nor WISKI will store user passwords.

I cannot log in to the Keycloak admin console, or configuration changes cannot be saved, and there is no clear error.

Keycloak is only compatible with the latest browser versions. Ensure your browser is up-to-date, this helped in a lot of cases.

Index

A

Access token	134
Application Metrics	195

B

Background processes	175
----------------------	-----

C

Client	49
CPU usage	176

D

Data Harvester	65
Data Viewer	199
Database	33, 34, 42
Database server	33
Dispatcher	8

H

here	76
------	----

I

Installation	
Client	28
Database	31
Initial	33, 100
Manually	100, 113
Requirements	12
Server	12
Silent mode	208
Storage	31
Update	113

J

java.exe	210
----------	-----

K

Keycloak	61, 131, 134
Keystore	61
KiDSM	200
KiGetMetrics	195
KiTSM	8

L

LDAP	131
Server configuration	132
LDAP attribute	134

Linux	121
-------	-----

M

Memory	176
Microsoft SQL Server	34
MitreID	131
MSSQL	134

O

ODBC	164, 165
ORACLE	33, 134

P

Package selection	10
Packages	10
Performance issue	174
Perl	46
PostgreSQL	42
Process Analytics	197

Q

Queues	177
--------	-----

R

RDP	174
Refresh token	134

S

Satellite node	228
Satellite server	115
Services	206
Silent mode	208
System architecture	8

T

TCA	9
Token claim name	134
Troubleshooting	159
TS Data Validation	203
TSM	8

U

User federation mapper	134
User federation provider	134
User model attribute	134

W

WDP	181
Windows 2022	174
WISKI Data Provider	181

WISKI server

46