

upKeeper Manager 6.0

Installation instructions

Copyright upKeeper Solutions AB

Revision 1.0

2026-05-18

Table of contents

Prerequisites.....	3
Report management	3
System overview	4
Installing Server Components	5
Installation – upKeeper Database	5
Installation – upKeeper Admin API	6
Installation – upKeeper Administration Website.....	12
Installation – upKeeper Client API.....	18
Configuring - upKeeper Admin API.....	24
Configuring - upKeeper Admin Web	27
Configuring - upKeeper Client API.....	29
Test – upKeeper Admin Web.....	31
Troubleshooting	32
Installation – upKeeper Application Server.....	33
Installation – upKeeper Client	41
Installation – upKeeper Client (silent).....	48
Installation - upKeeper Files Website.....	51
Build upKeeper DSOS or SOS.....	58
Install Microsoft software	58
Add upKeeper DSOS or SOS files	58
Configure upKeeper SOS	59
Update upKeeper DSOS or SOS	60
Diagnostics in upKeeper DSOS	60
Configuration - upKeeper 5.X.....	61
Configuration – Organization settings.....	65

Appendix A - Configuration of Windows Deployment Services 69

Appendix B – upKeeper white label 71

Appendix C – Set up Singel Sign-On 73

Appendif D - Custom script during Windows Setup..... 75

Prerequisites

Prerequisites are described in the document “upKeeper 6.0 - Installation Prerequisites”.

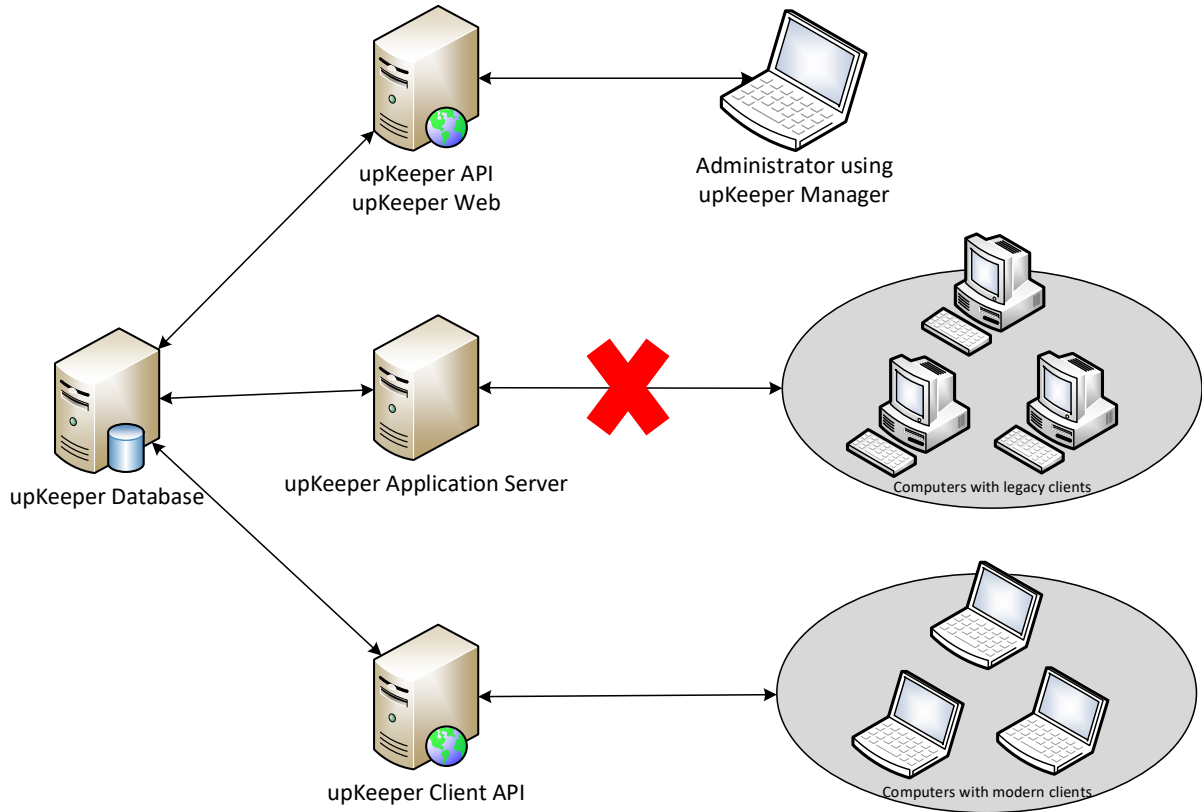
Report management

Reports are created for Microsoft Reporting Services and they can be found in “upKeeper Reports 5.2.13.2.zip” file or later version. Reports can be uploaded and configured manually one by one or by using report upload script found in “ReportUploadScript.zip”. Report upload script zip-file include instructions.

System overview

Server components can all be installed on the same server or spread on different servers. If you are new to the system or want advice, please consult with an upKeeper expert.

Notice! Older upKeeper Manager Clients and upKeeper Manager SOS that requires WCF (Windows Communication Framework) is no longer supported. upKeeper Manager Clients and upKeeper Manager SOS must be of upKeeper version 4.0 or newer.



Installing Server Components

Note that before installing, you must notify your upKeeper reseller or upKeeper Solutions to activate the license.

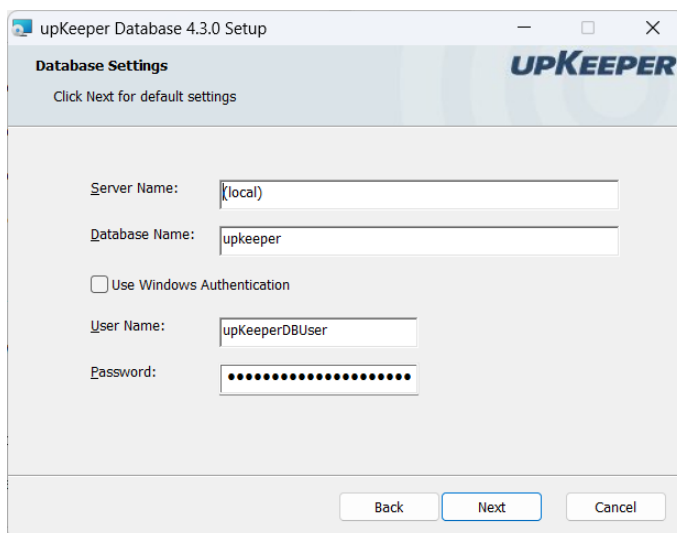
Installation – upKeeper Database

- Sign in with administrative rights to the server that will be used for the upKeeper database.
- Restore clean database backup file from upKeeper Solutions SFTP or execute the installation file “upKeeper.DB.4.3.0.msi” and upgrade to correct version.

Using MSI for database installation

- Select server\instance.
- Enter the name of the upKeeper database.
- Specify if you want to use Integrated Security or SQL Server Authentication.

Note! The rest of this document will assume that SQL Server Authentication is used



The screenshot shows the 'upKeeper Database 4.3.0 Setup' window with the 'Database Settings' tab selected. The window title bar includes the text 'upKeeper Database 4.3.0 Setup' and standard window controls. The 'Database Settings' section has a sub-header 'UPKEEPER' and a note 'Click Next for default settings'. The form contains the following fields and options:

- Server Name:** A text box containing '(local)'. The label has a small 'S' underlined.
- Database Name:** A text box containing 'upkeeper'. The label has a small 'D' underlined.
- Use Windows Authentication:** An unchecked checkbox.
- User Name:** A text box containing 'upKeeperDBUser'. The label has a small 'U' underlined.
- Password:** A password field with 12 dots. The label has a small 'P' underlined.

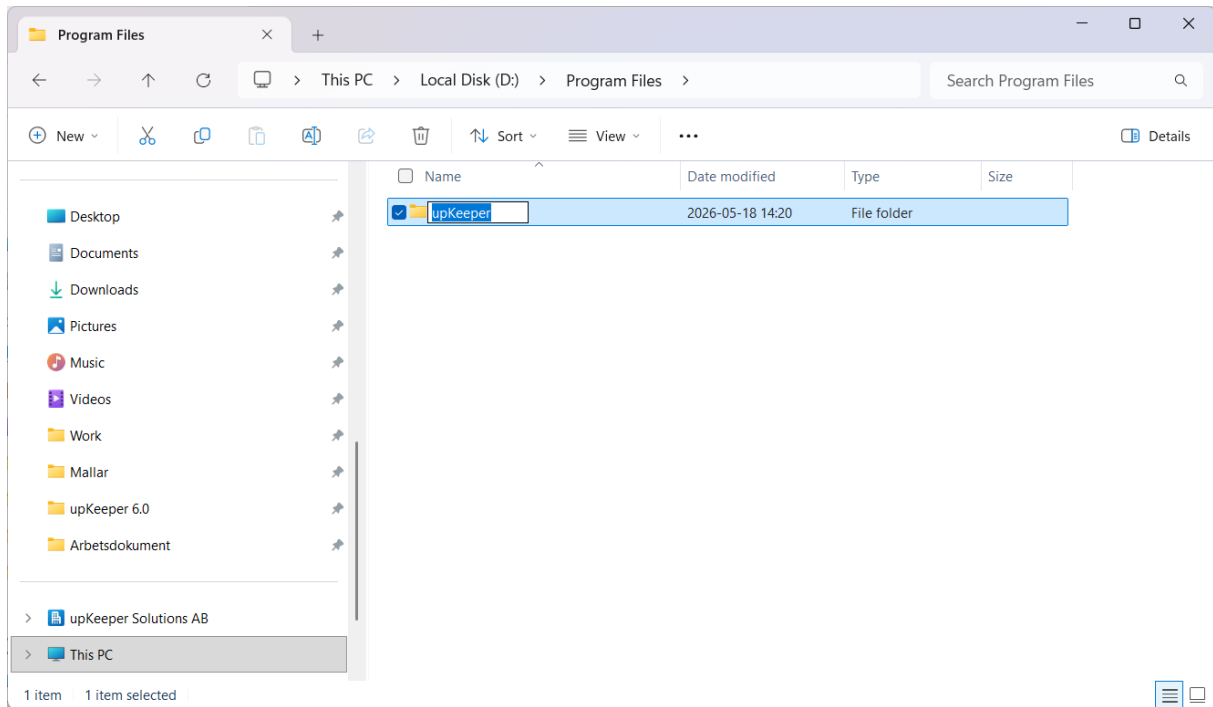
At the bottom of the window, there are three buttons: 'Back', 'Next', and 'Cancel'. The 'Next' button is highlighted with a blue border.

- If necessary, run database update script to reach current version. Check every execution before running next. Database backup between versions upgrades is recommended.

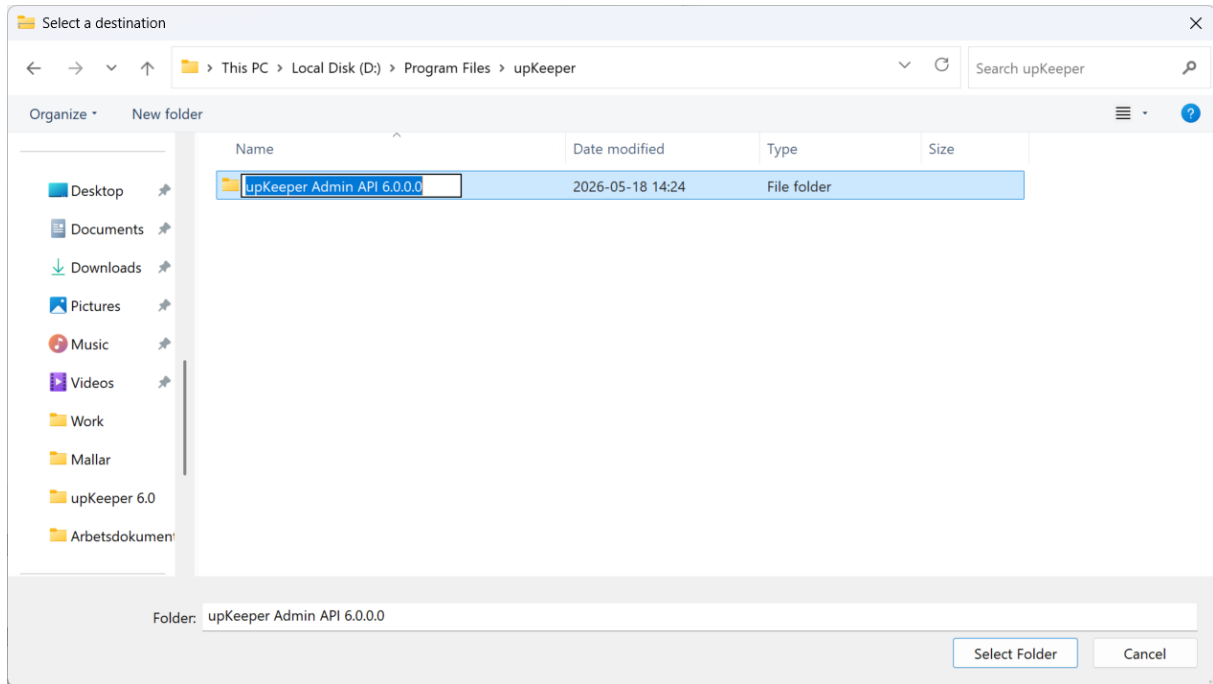
Installation – upKeeper Admin API

Sign in with administrative rights to the server that will be used for the upKeeper Admin API.

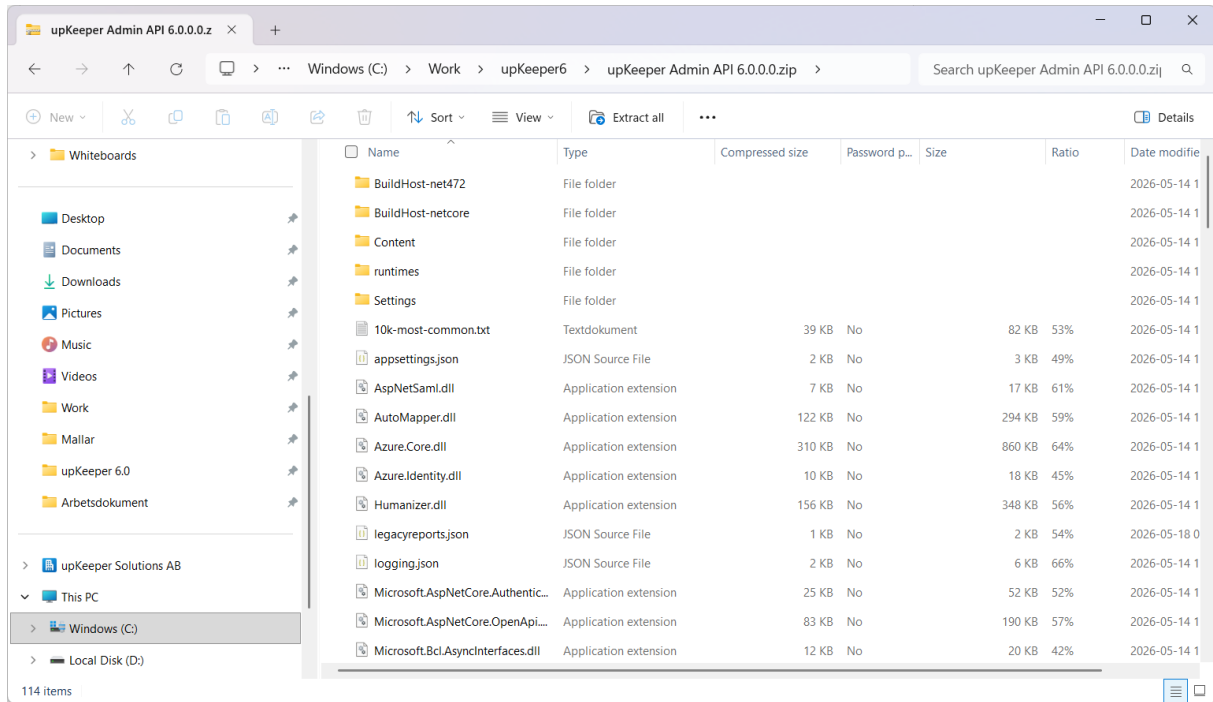
Create a folder named “upKeeper” in the path “C:\Program Files\”.



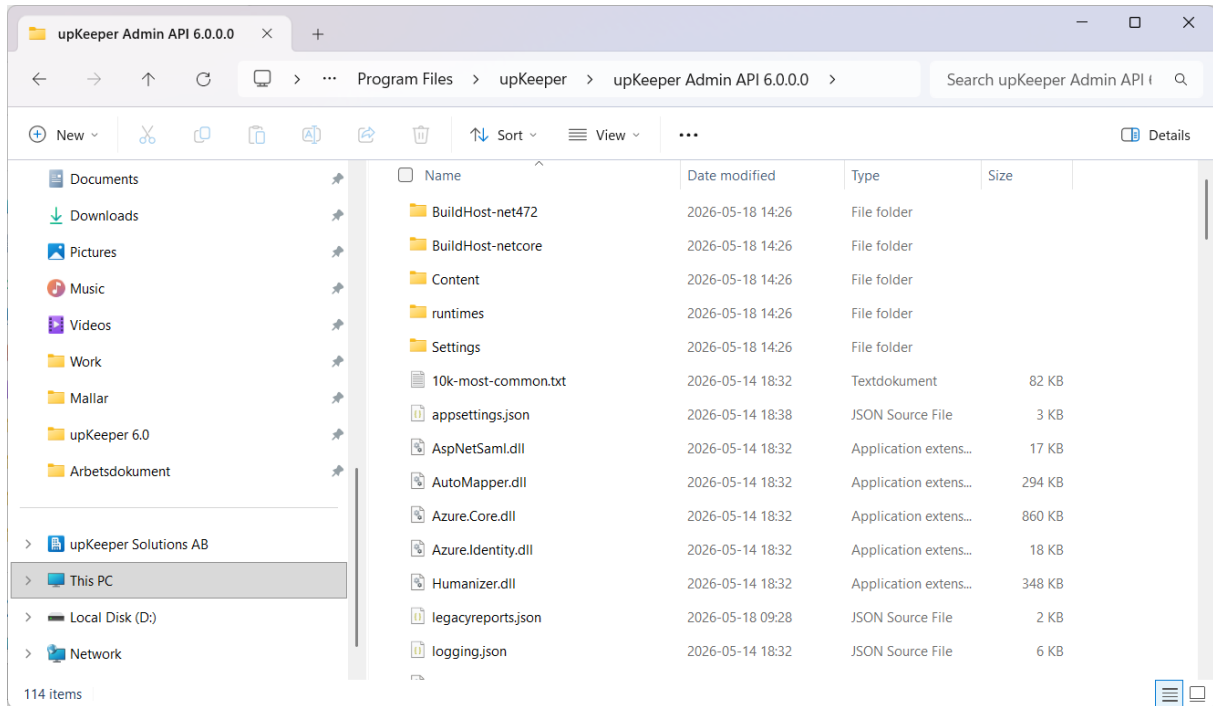
Create a folder named according to the zip-file name (ex upKeeper Admin API 6.0.0.0) in the path “C:\Program Files\upKeeper”.



Extract the content of the file “upKeeper Admin API 6.x.x.zip” into the folder just created.

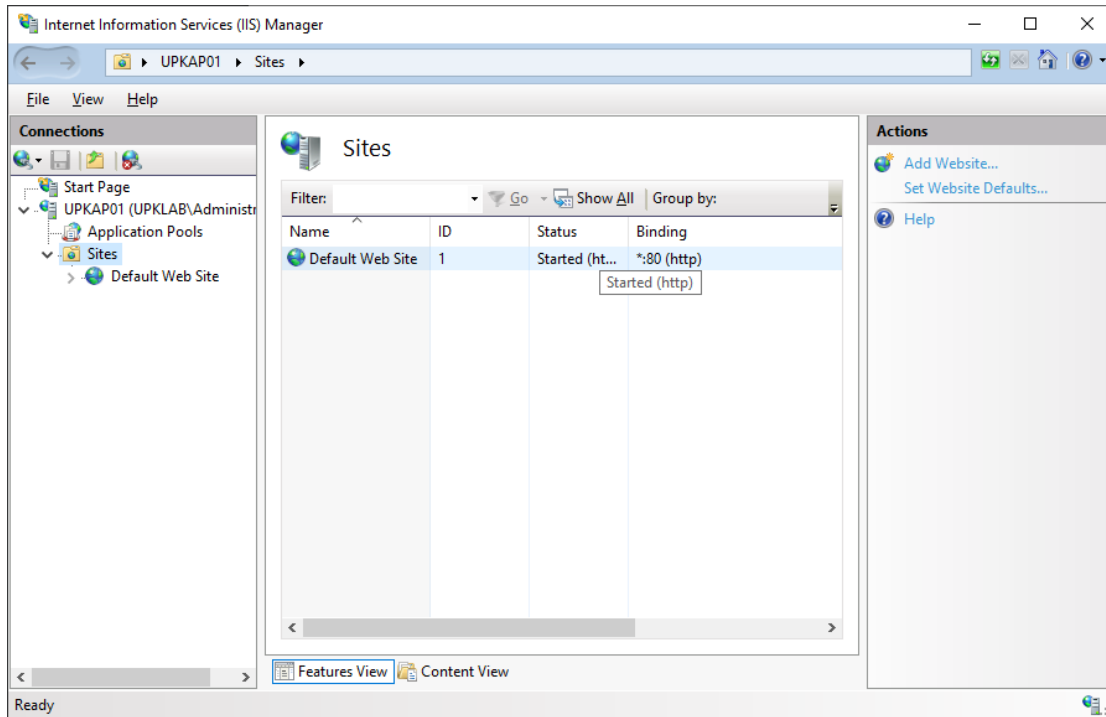


The upKeeper Admin API folder after extraction (content can look different due to version).



Add folder "log" and assign local group IIS_IUSRS rights to modify.

Open "Internet Information Services (IIS) Manager" on the server.



Add new web site in “Internet Information Services (IIS) Manager”. The “Physical path” should point to the folder where you extracted the API files. Remember to change the “host name” to a DNS address reachable from where the system will be administrated.

Add Website ? X

Site name: Application pool:

Content Directory

Physical path:

Pass-through authentication

Binding

Type: IP address: Port:

Host name:

Require Server Name Indication

Disable TLS 1.3 over TCP Disable QUIC

Disable Legacy TLS Disable HTTP/2

Disable OCSP Stapling

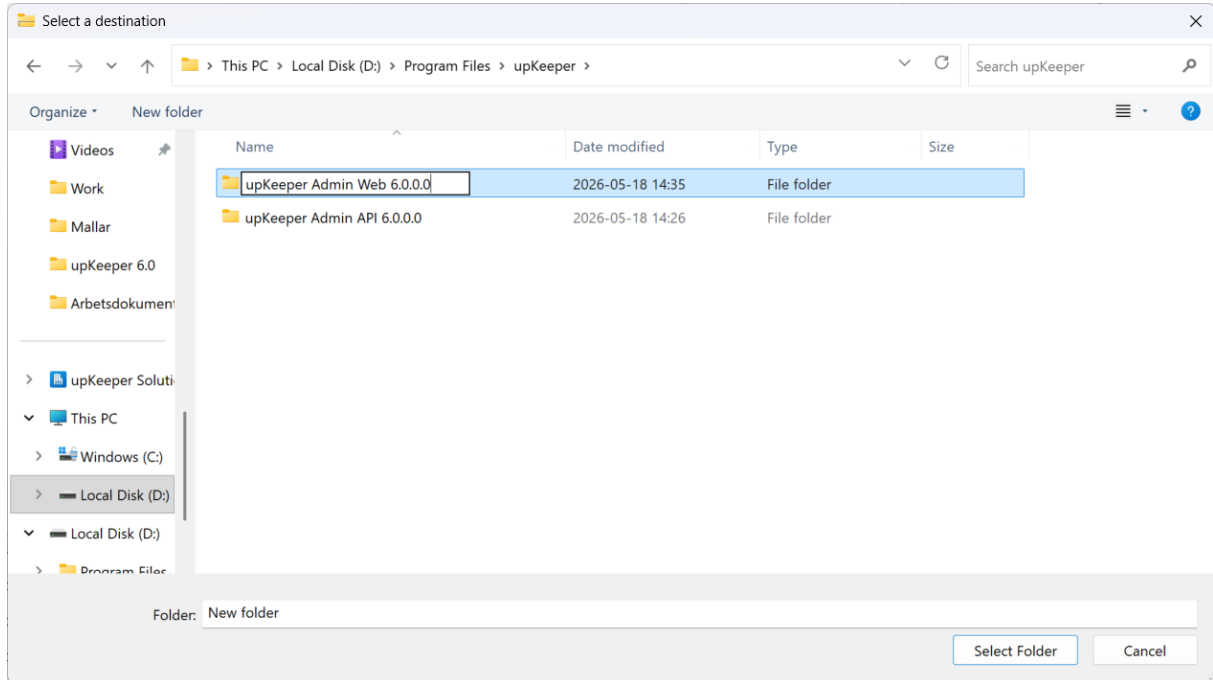
SSL certificate:

Start Website immediately

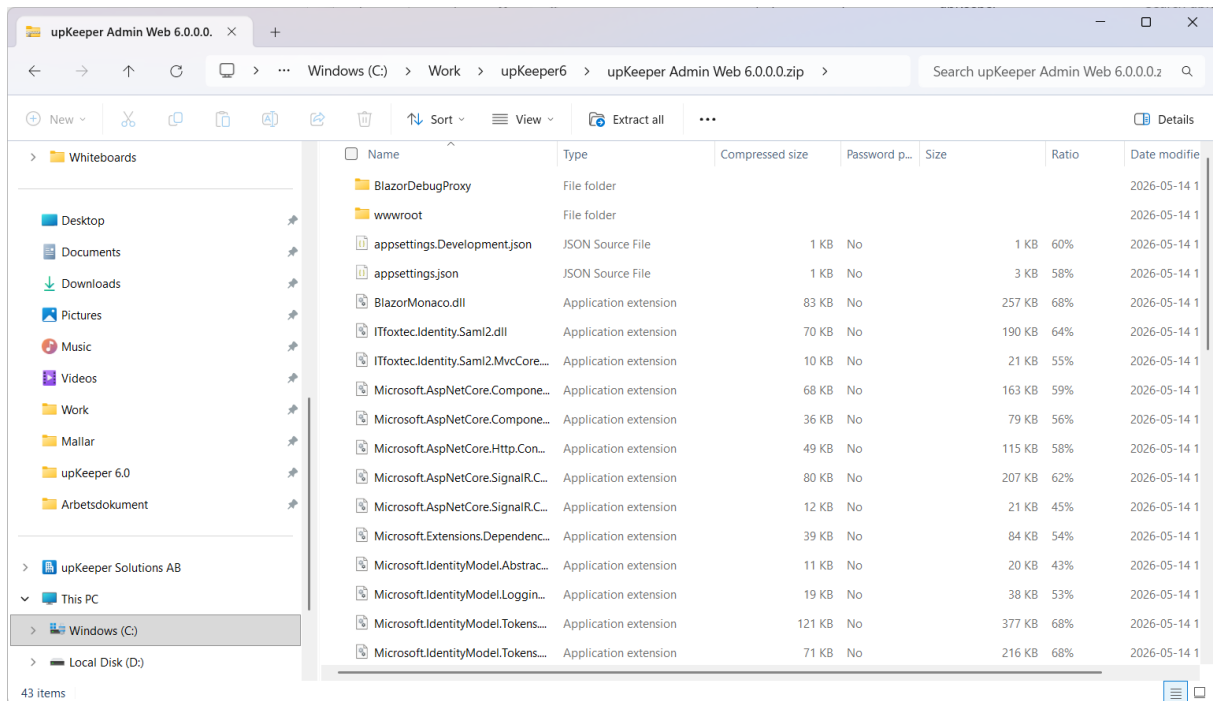
Installation – upKeeper Administration Website

Sign in with administrative rights to the server you used for the upKeeper API Website

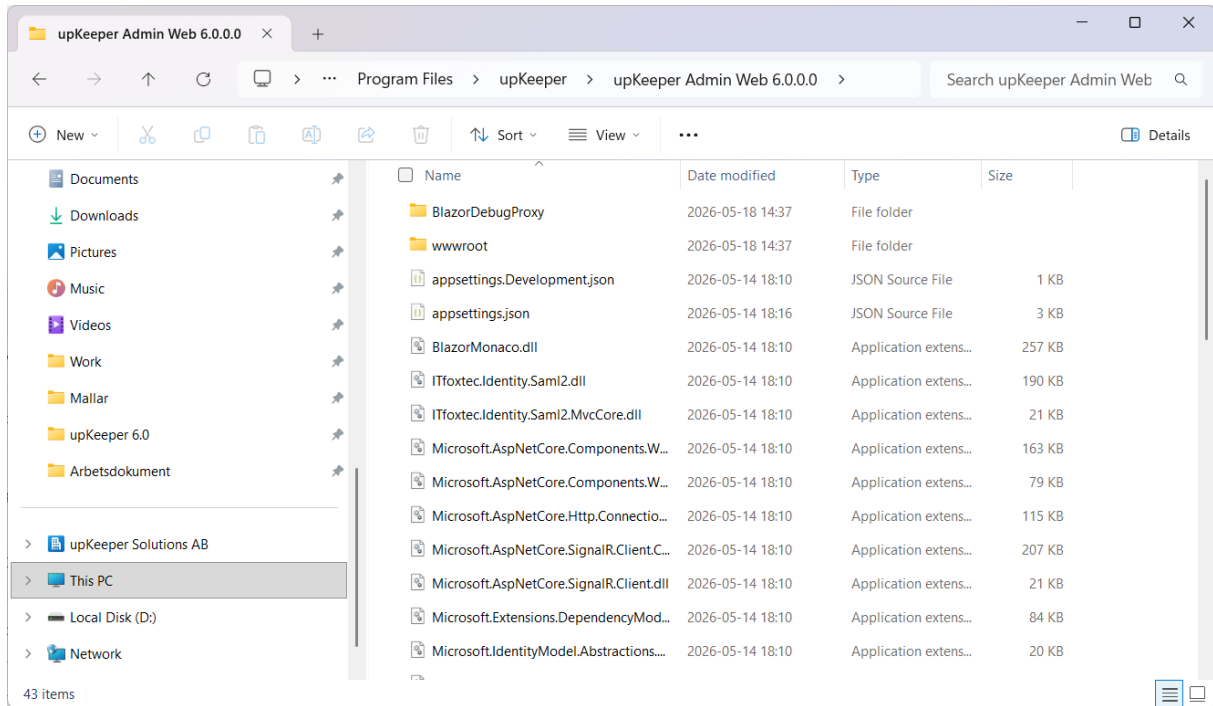
Create a folder named according to current upKeeper Admin Web zip-file in the path “C:\Program Files\upKeeper”.



Extract the content of the file “upKeeper Admin Web 6.x.x.zip” into the “upKeeper Admin Web 6.x.x” folder just created.

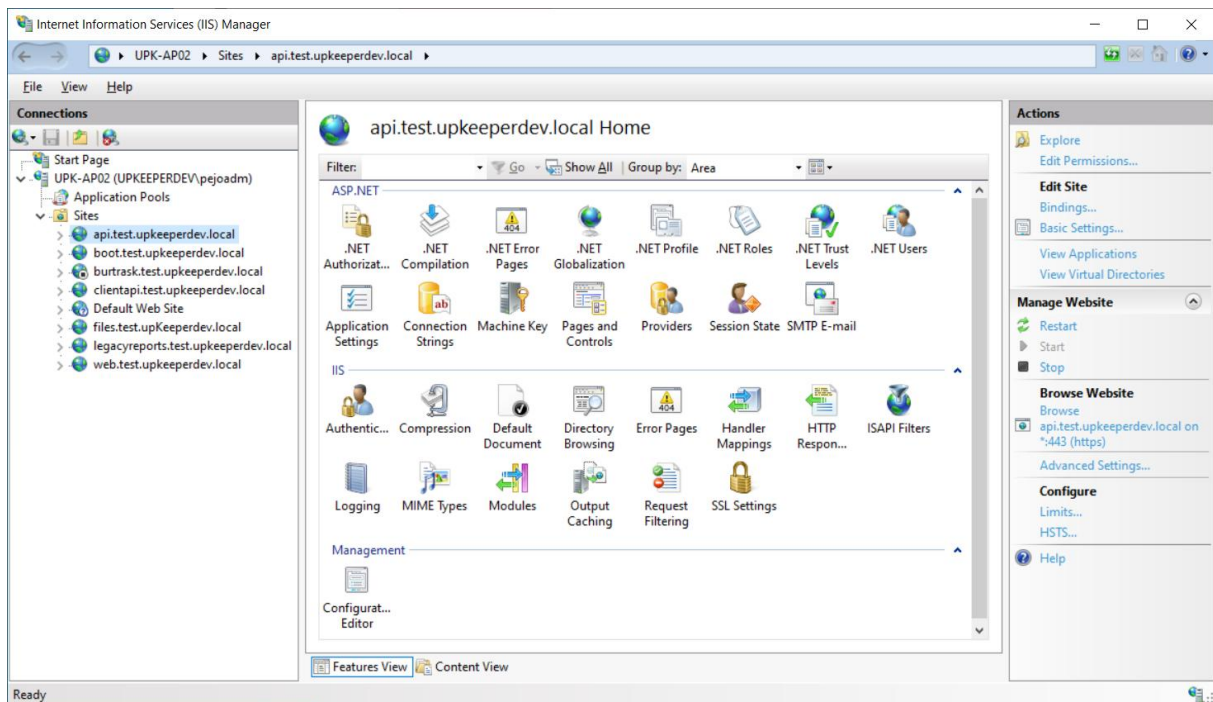


The upKeeper Admin web folder after extraction (content can look different due to version).



Add folders “log” and “dpkeys” and assign local group IIS_IUSRS modify rights.

Open "Internet Information Services (IIS) Manager" on the server



Add new web site in "Internet Information Services (IIS) Manager". The "Physical path" should point to the folder where you extracted the web files. Remember to change the "host name" to a DNS address reachable from where the system will be administrated.

Add Website ? X

Site name: Application pool:

Content Directory

Physical path:

Pass-through authentication

Binding

Type: IP address: Port:

Host name:

Require Server Name Indication

Disable TLS 1.3 over TCP Disable QUIC

Disable Legacy TLS Disable HTTP/2

Disable OCSP Stapling

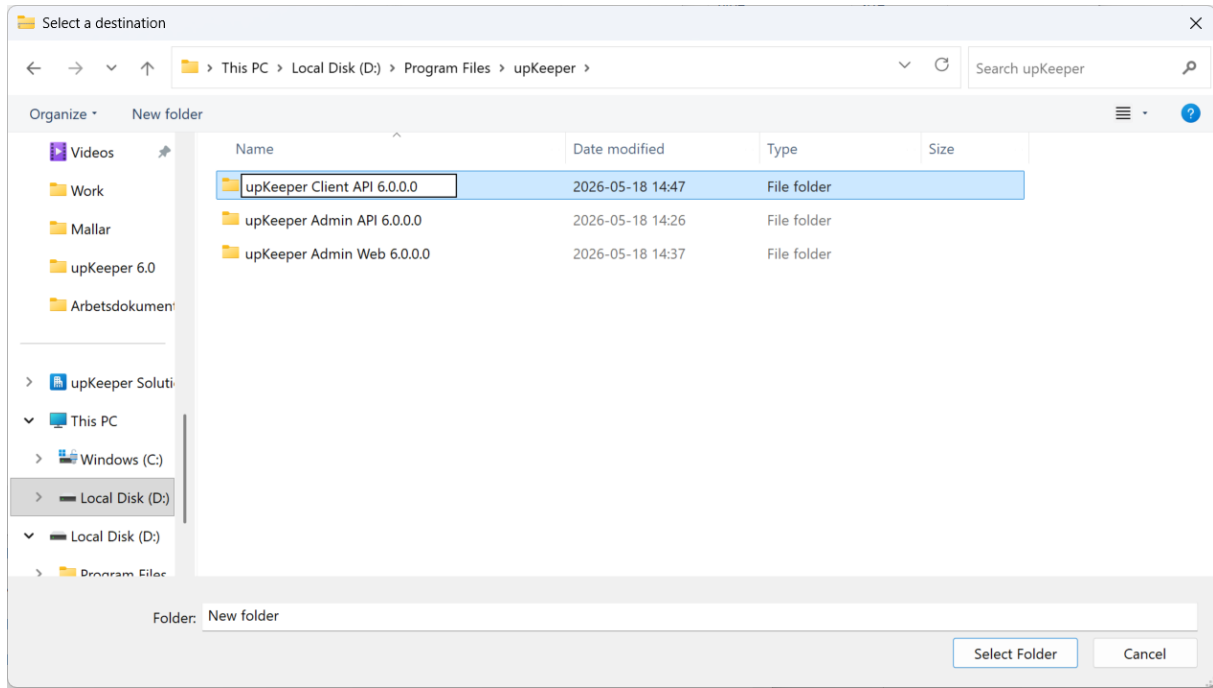
SSL certificate:

Start Website immediately

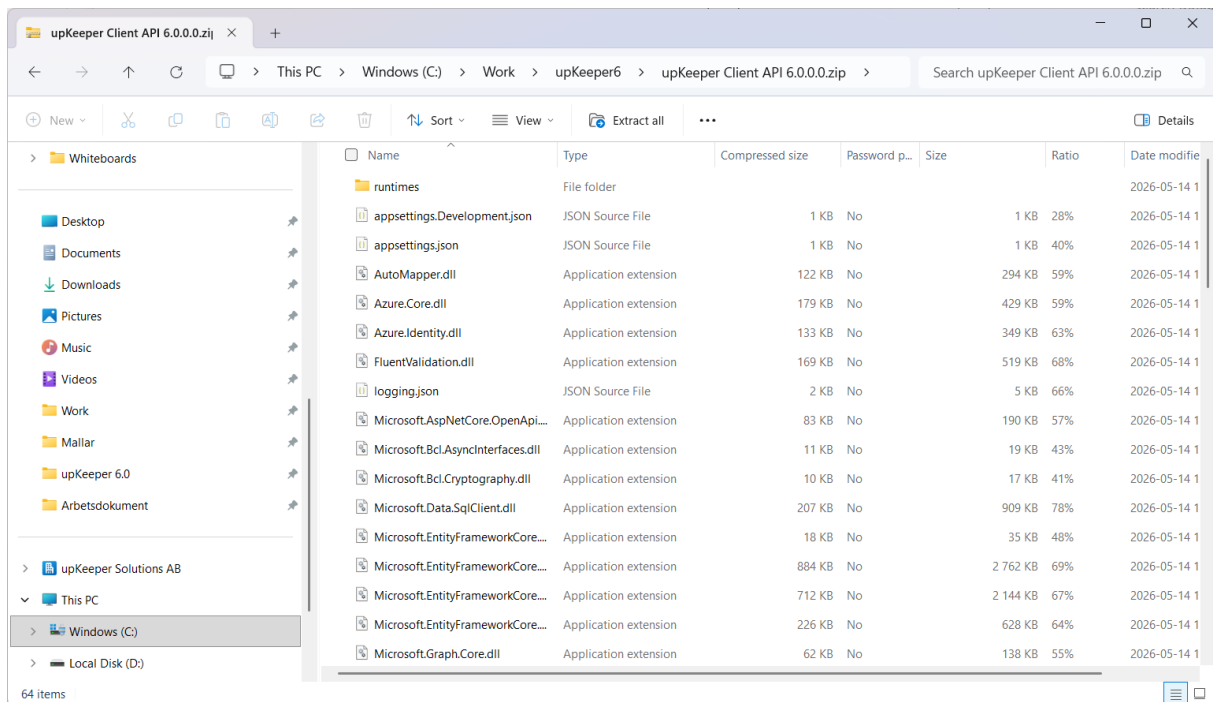
Installation – upKeeper Client API

Sign in with administrative rights to the server you used for the upKeeper Administration Website or any server with the feature Internet Information Server enabled.

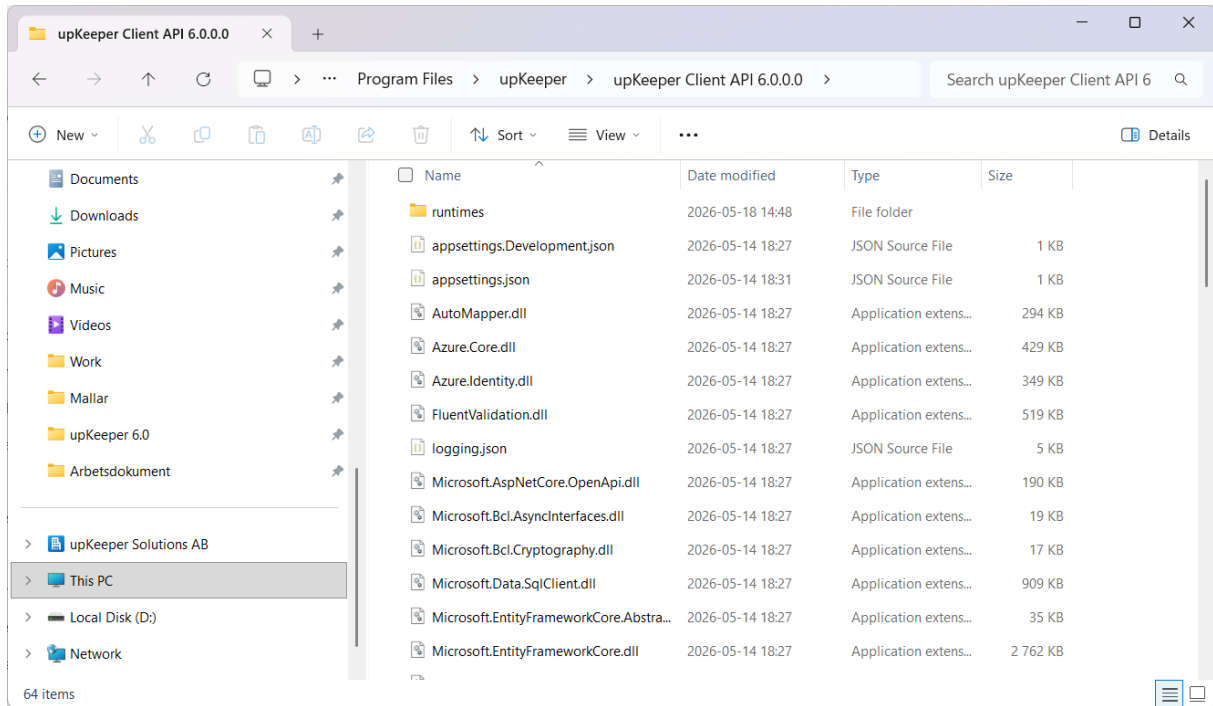
Create a folder according to the name of the zip file, in the path “C:\Program Files\upKeeper”.



Extract the content of the file “upKeeper Client API 6.x.x.zip” into the folder just created.

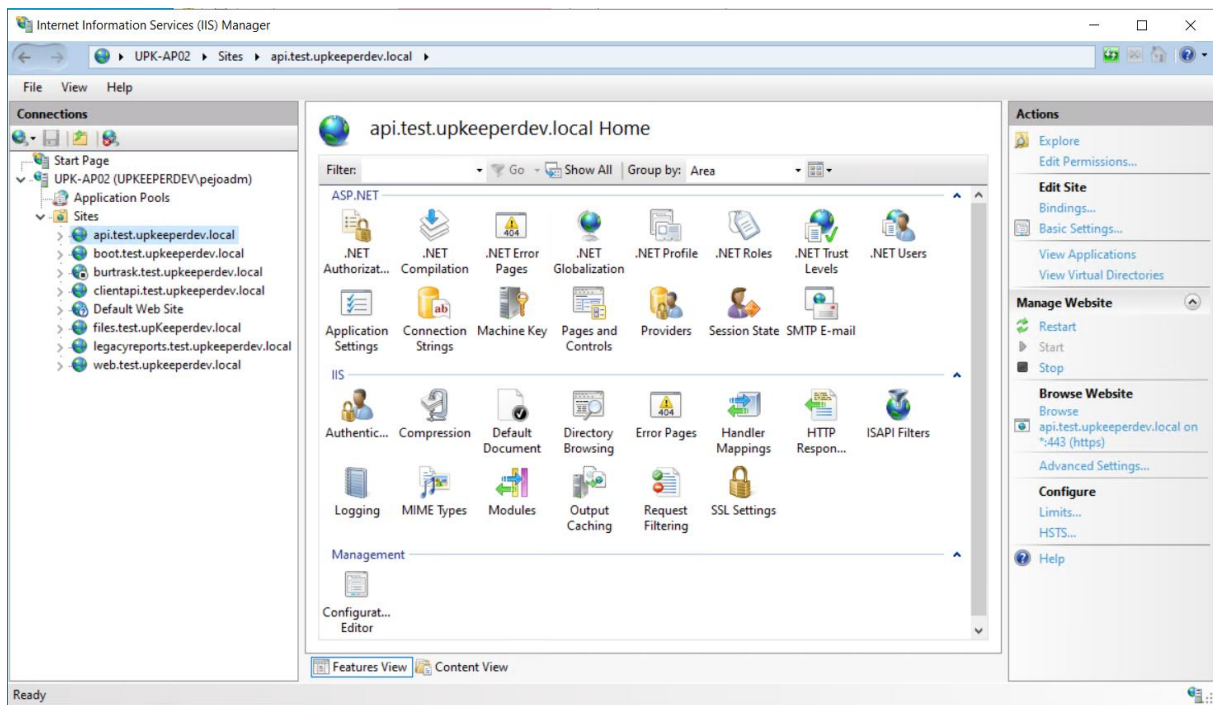


The upKeeper Client API folder after extraction. (Content can look different due to version)



Add folder "log" and assign local group IIS_IUSRS modify rights.

Open "Internet Information Services (IIS) Manager" on the server



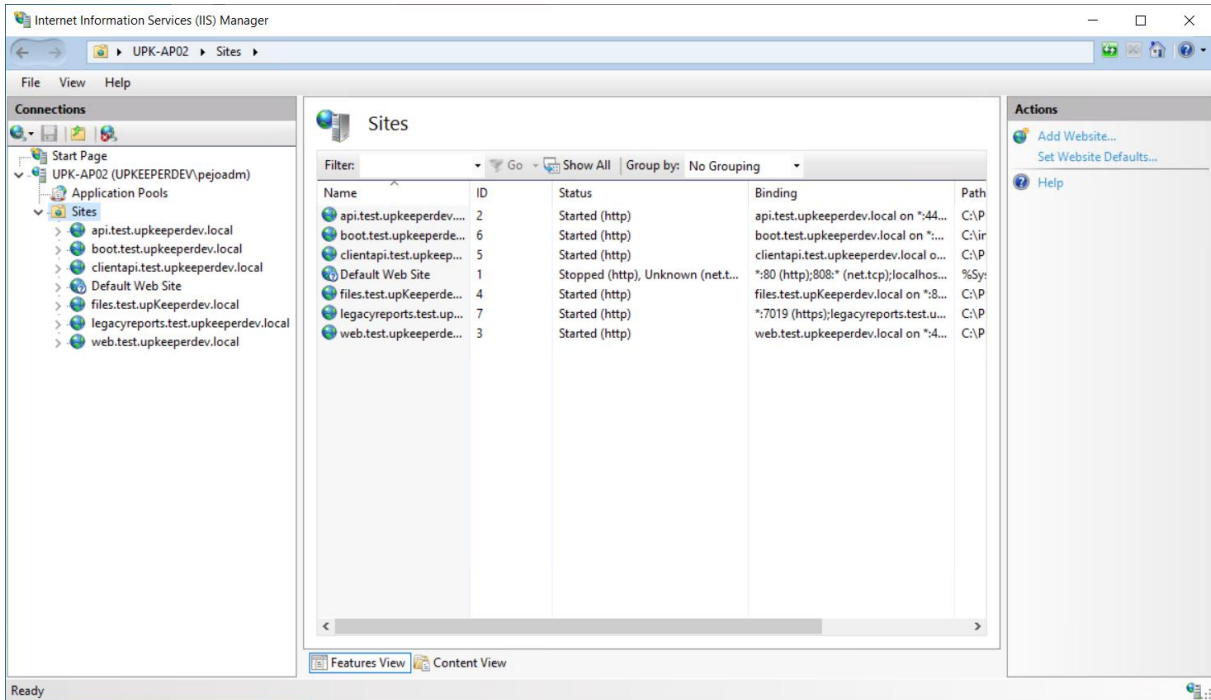
Add new web site in “Internet Information Services (IIS) Manager”. The “Physical path” should point to the folder where you extracted the web files. Remember to change the “host name” to a DNS address reachable from the clients.

The screenshot shows the 'Add Website' dialog box in IIS Manager. The fields are filled as follows:

- Site name:** clientapi.test.upkeeperdev.local
- Application pool:** clientapi.test.upkeeperdev.local
- Content Directory:**
 - Physical path:** C:\Program Files\upkeeper\upKeeper Client API 6.0.0.0
 - Pass-through authentication:** Connect as... Test Settings...
- Binding:**
 - Type:** https
 - IP address:** All Unassigned
 - Port:** 443
 - Host name:** clientapi.test.upkeeperdev.local
 - Require Server Name Indication
 - Disable TLS 1.3 over TCP
 - Disable Legacy TLS
 - Disable OCSP Stapling
 - Disable QUIC
 - Disable HTTP/2
- SSL certificate:** *.test.upkeeperdev.local
- Start Website immediately

Buttons: OK, Cancel

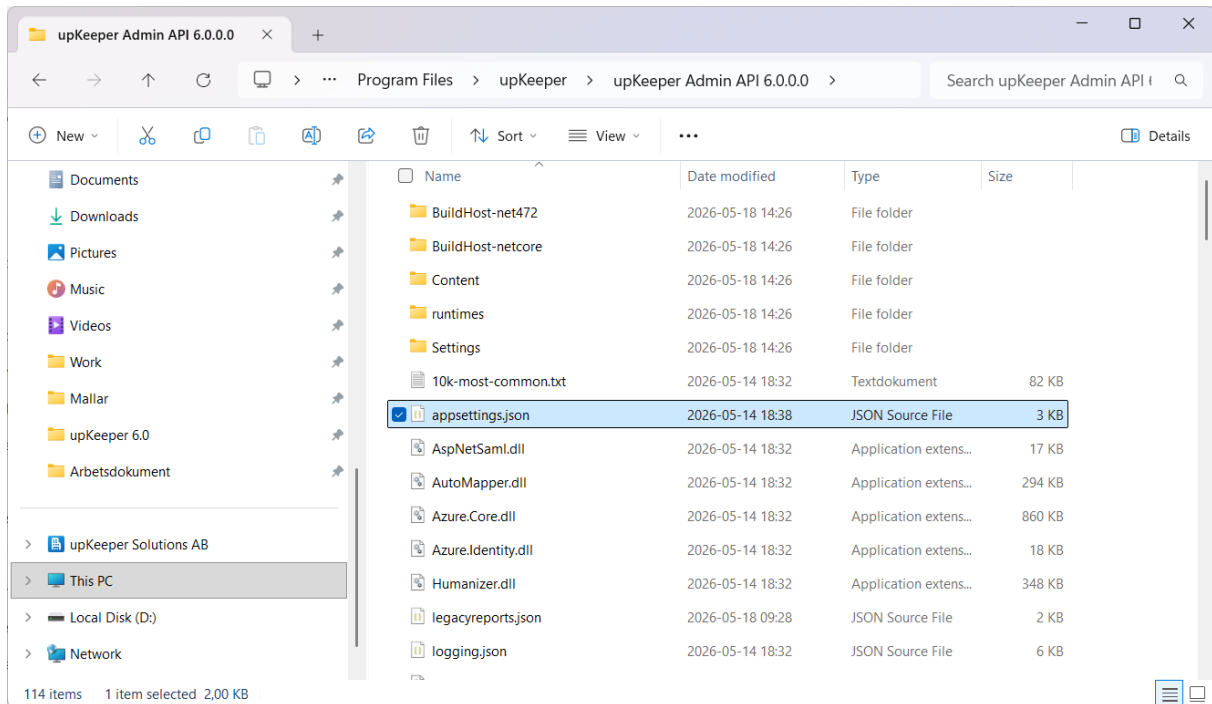
Available web sites should look like this when all components is installed on one server. This server has upKeeper Admin API, upKeeper Admin Web, upKeeper Client API, upKeeper Files, upKeeper Legacy Reports and upKeeper PXE over HTTP boot (testing).



Configuring - upKeeper Admin API

Edit the configuration file

Perform the following changes in the configuration files (appsettings.json, logging.json) located in the **Destination Folder** (C:\Program Files\upKeeper\upKeeper Admin API 6.0.0.0) specified during the installation.



Database settings

Database connection settings is configured in appsettings.json file under connection strings.

```
"ConnectionStrings": {
  "UpkeeperDb": "Data Source=[DATABASE_SERVER];Initial Catalog=[DATABASE];User
ID=[DATABASE_USER];Password=[DATABASE_PASSWORD];Integrated
Security=False;MultipleActiveResultSets=True;"
  "WSUSBaseUrlApi": ""
}
```

Note! The text in the above example has extra line breaks that are not allowed in the configuration file. Backslash is a reserved character and must be replaced with double backslash.

- **Data Source** = the address of your database server.
- **Initial Catalog** = database name
- **User ID** = the name of the database user that owns the upKeeper database. Replace [USERNAME] with sql user.
- **Password** = the password for the above user. Replace [PASSWORD] with password for the sql user.
- **Integrated Security** = true mean that user browsing the website will be used to connect to the upkeeper database. If specified User ID should be used you need to set this property to false.

Database settings for logging

Logging is set in the file **logging.json**. Locate the following tags:

```
"UpkeeperDb": {
  "type": "Database", "connectionString": "Data Source=[DATABASE_SERVER];Initial
Catalog=[DATABASE];User ID=[DATABASE_USER];Password=[DATABASE_PASSWORD];Integrated
Security=False;MultipleActiveResultSets=True;"
```

Edit the connection string properties so that they have the same values as the connection string in the previous section.

Level of logging is changed under the rules section. Approved values are ERROR, WARN, INFO and DEBUG. In production we recommend WARN, but for test or troubleshoot you should use higher level of logging like INFO or DEBUG. Details can be found inside the file.

Reporting Services settings

Report settings is set in the **appsettings.json** file. Edit these if you are to use upKeeper reporting.

Locate the tag **ReportingServiceClientOptions**:

```
"ReportingServiceClientOptions": {
  "BaseUrl": "https://[REPORTSERVER_URL]/",
  "Username": "[REPORTSERVER_USERNAME]",
  "Password": "[REPORTSERVER_PASSWORD]",
  "Domain": "[DOMAIN]",
  "LinkToReportingServer": false
}
```

- **BaseURL** = the address of your SQL Server Reporting Services installation (you will find the name if you open the SQL Reporting Services Configuration Manager, Web Portal URL)
- **Username** = the username of an account that has the right to run reports
- **Password** = the password for the account above
- **Domain** = domain for user account specified.
- **LinkToReportingServer** = Specify if report server should be used directly and not thru upKeeper Manager.

WSUS settings

Locate the tag **ConnectionStrings**.

```
"WSUSBaseUrlApi": "[WSUS_SERVER]:[WSUS_SERVER_PORT]"
```

- **WSUSBaseUrlApi** = address where upKeeper WSUS service is installed and which port is used. WSUS uses port 9001 as default.

CORS

To allow web to connect with the API you must allow that address.

```
"Cors": [
  "https://localhost",
  "https://[UPKEEPER_ADMIN_WEB_URL]"
```

Cors = URL address to your admin web. Multiple address can be specified separated with comma (,).

Two factor authentication

Locate tag UpkeeperSettings and edit if you want to use two factor authentication. Note! Before activating you should specify service users by enabling that parameter in the users view.

```
"UpkeeperSettings": {
  "ComputerLogFileFolder": "C:\\temp\\filelogs",
  "AllowAddToLockedObjects": false,
  "UseTFA": true,
  "AllowEmailTFA": false,
  "TFAMailFrom": "noreply@yourcompany.local",
  "TFAMailSubject": "upKeeper Manager PIN code", "PasswordPolicy": {
    "RegExp": "^(?=.*[A-Z])(?=.*[a-z])(?=.*[!@#%&*()])?(?=.*[0-9]).{15,}$",
    "DisallowedPasswordListFileLocation": "10k-most-common.txt"
  },
  "AutopilotFactoryResetKeepEnrollmentData": false,
  "AutopilotFactoryResetKeepUserData": false,
  "PublicKey": {
    "Modulus":
"0oiiOH08WTZSFRAz0398qR++SgLiDCR8BimYYopHkaepstTeyiBU+YHVWHPJuk/CTbfIfadahZugzTcxVihNU5PusRZf36fXTZxpcV
OhV9o7czktJrEOjYVvAs2Rn+bMPe+DNS0zLIpR+peIj+Tuomz6PgsIGxLu/riBMTXl4HE=",
    "Exponent": "AQAB"
  },
  "InstallationId": "3a33bd45-ce0b-4644-896b-bd1350ace001",
  "SMTP_Username": "[Username]",
  "SMTP_Password": "[Password]",
  "SMTP_Host": "HOST",
  "SMTP_Port": 587,
  "SMTP_UseSSL": true
},
```

- ComputerLogFileFolder = Folder where files sent from client thru logging function will be saved.
- UseTFA = Set to true to enable two factor authentication.
- AllowEmailTFA = Set to true if users should be able to verify login with email.
- TFAMailFrom = Address used when sending authentication emails.
- TFAMailSubject = Subject of email sent for two factor authentication.
- SMTP_Username = Username of user with access to configured SMTP address (Address is configured in web.config).
- SMTP_Password = Password for SMTP username.

Other Settings

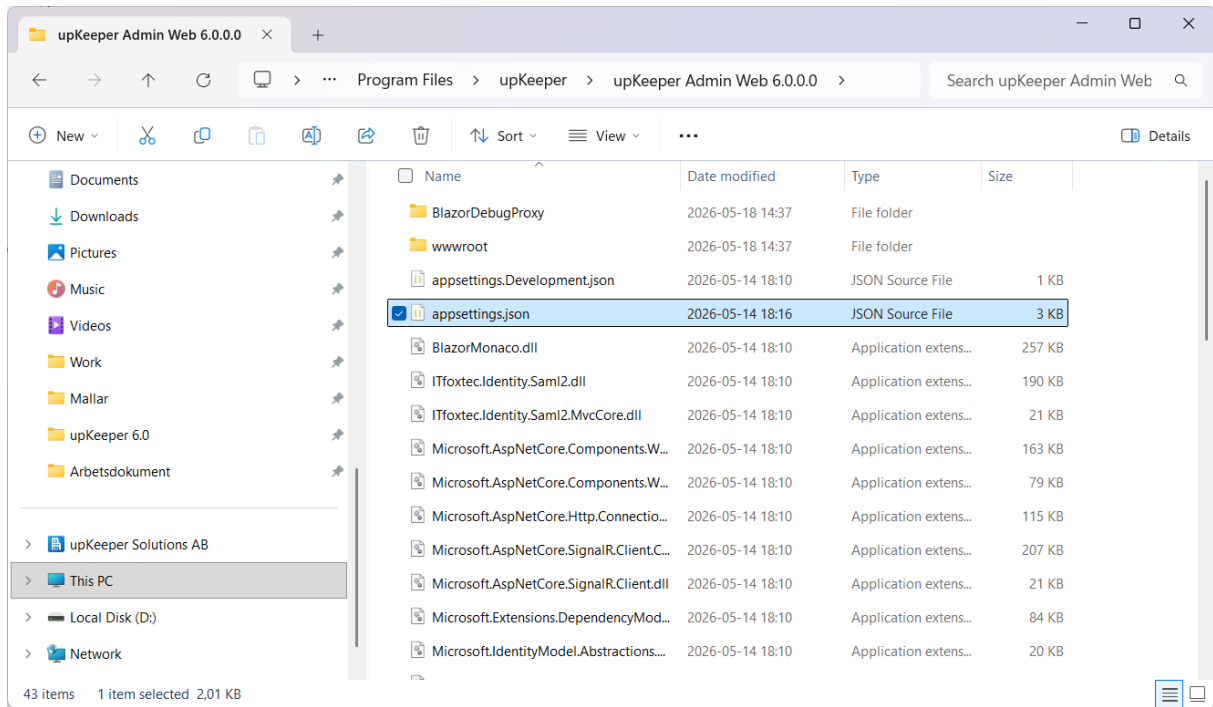
- InstallationId = Value to identify a specific API if you have multiple.
- RefreshTokenExpiryInDays = Default is 7 days. This value defines the expire time of the access token to the API. When expiring a refresh token will be used to get a new one.
- MinPasswordLength = Default is 5 characters. This value defines the minimum number of characters a password can include.
- PublicKey.Modulus and PublicKey.Exponent = Used to read central encrypted information. Values should not be changed.

Settings not documented are settings that should not be changed unless advised by upKeeper personnel.

Configuring - upKeeper Admin Web

Edit the configuration file

Perform the following changes in the file **appsettings.json**, located in the **Destination Folder** (C:\Program Files\upKeeper\upKeeper Admin Web 6.0.0.0) specified during the installation.



```

D:\Program Files\upKeeper\upKeeper Admin Web 6.0.0.0\appsettings.json - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
appsettings.json
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning",
6       "Microsoft.AspNetCore.Antiforgery": "Debug"
7     }
8   },
9   "Serilog": {
10    "Using": [ "Serilog.Sinks.File", "Serilog.Enrichers.Environment" ],
11    "MinimumLevel": {
12      "Default": "Information",
13      "Override": {
14        "Microsoft": "Warning",
15        "Microsoft.AspNetCore": "Warning",
16        "System": "Warning"
17      }
18    },
19    "Enrich": [ "FromLogContext", "WithMachineName", "WithProcessId", "WithThreadId" ],
20    "WriteTo": [
21      {
22        "Name": "Console",
23        "Args": {
24          "outputTemplate": "[{Timestamp:HH:mm:ss} {Level:u3}] {SourceContext} {Message:l} {NewLine}{Exception}"
25        }
26      },
27      {
28        "Name": "File",
29        "Args": {
30          "path": "Logs/upkeeper-adminweb-.log",
31          "rollingInterval": "Day",
32          "fileSizeLimitBytes": 10485760,
33          "rollOnFileSizeLimit": true,
34          "retainedFileCountLimit": 14,
35          "shared": true,
36          "flushToDiskInterval": 1,
37          "outputTemplate": "{Timestamp:yyyy-MM-dd HH:mm:ss.fff} [{Level:u3}] {SourceContext} {Message:l} {Propertie
38        }
39      }
40    ]
41  },
42  "AllowedHosts": "*",
43  "UpKeeper": {
44    "ServerUrl": "https://[ADMIN_WEB_URL]/",
45    "ApiServerUrl": "https://[ADMIN_API_URL]/",
46    "ApiBaseUrl": "https://[ADMIN_API_URL]/api/",
47    "ApiCertificateThumbprints": "",
48    "Saml": {
49      "SpSigningCertificateThumbprint": "4c9ac61c781876cf3743bcd0aa38ddcded034875"
50    }
51  },
52  "Edition": {
53    "Id": "Standard",
54    "EnabledFeatures": "Applications.View, Computers.AllTabs, Users.View, Groups.View, Hardware.ApplicationTab, Distr
55  },
56  "Zendesk": {
57    "WidgetKey": "68ca3700-f1a5-4427-90a0-354d33f4c10a"
58  }
59 }
60
JSON file length: 2 059 lines: 60 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 IN

```

```

"AllowedHosts": "*",
"UpKeeper": {
  "ServerUrl": "https://[ADMIN_WEB_URL]/",
  "ApiServerUrl": "https://[ADMIN_API_URL]/",
  "ApiBaseUrl": "https://[ADMIN_API_URL]/api/",
  "ApiCertificateThumbprints": "",
  "Saml": {
    "SpSigningCertificateThumbprint": "[CERTIFICATE_THUMBPRINT]"
  }
},

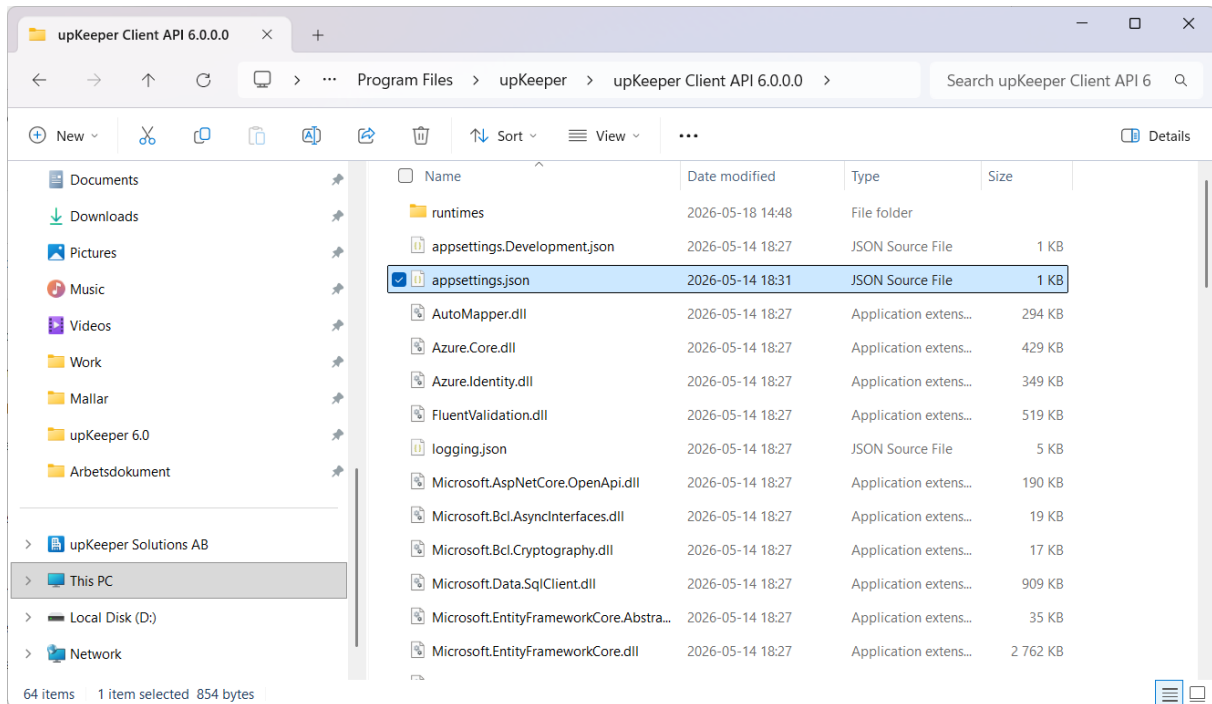
```

- **ServerUrl** – address of upKeeper Admin Web.
- **ApiServerURL** – the address of upKeeper Admin API.
- **ApiBaseUrl** – same base address as **APIServerUrl** with the addition of “api/”.
- **APICertificateThumbprints** – Used for self-signed certificates (test environment)
- **SpSigningCertificateTumbprint** – Thumbprint of certificate used for signing SAML requests.

Configuring - upKeeper Client API

Edit the configuration file

Perform the following changes in the file **appsettings.config**, located in the **Destination Folder** (C:\Program Files\upKeeper\upKeeper Client API 6.0.0.0) specified during the installation.



Database settings

Locate the tag **"ConnectionStrings"**.

```
"ConnectionStrings": {
  "UpkeeperDb": "Data Source=[DATABASE_SERVER];Initial Catalog=[DATABASE];User
ID=[DATABASE_USER];Password=[DATABASE_PASSWORD];Integrated
Security=False;MultipleActiveResultSets=True;TrustServerCertificate=True",
  "WSUSBaseUrlApi": ""
}
```

Note! The text in the above example has extra line breaks that are not allowed in the configuration file. Backslash is a reserved character and must be replaced with double backslash.

- **Data Source** = the address of your database server.
- **Initial Catalog** = database name
- **User ID** = the name of the database user that owns the upKeeper database. Replace [DATABASE_USER] with sql user.
- **Password** = the password for the above user. Replace [DATABASE_PASSWORD] with password for the sql user.
- **Integrated Security** = true mean that user browsing the website will be used to connect to the upkeeper database. If specified User ID should be used, you need to set this property to false.
- **TrustedServerCertificate** = Trust certificate used.

Logging

Low level logging can be enabled but is not recommended in a production environment. Changes can be made in logging config file named “**logging.json**”

Logging level can be changed to get more or less information. Approved log levels ERROR, WARN, INFO and DEBUG. We recommend log level WARN in production, but INFO or DEBUG can be used in test or troubleshoot situations. Details can be found inside the file.

upKeeper specific settings

Specific upKeeper settings can be changed and set in the file “**appsettings.json**”

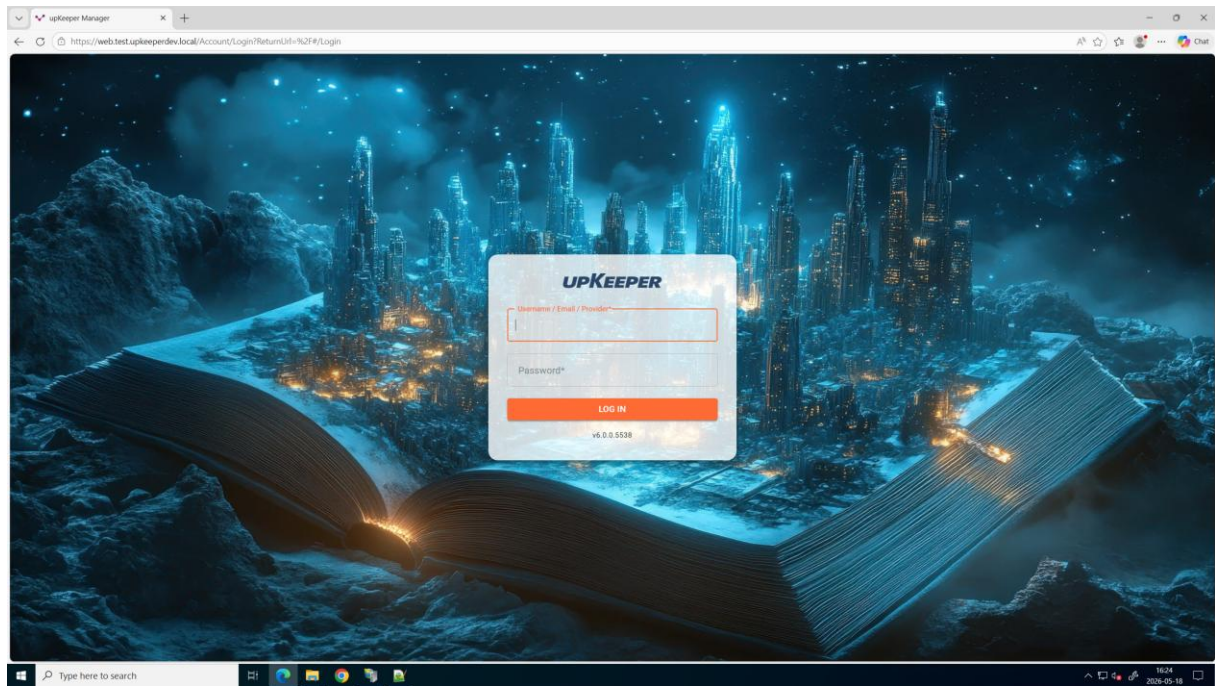
```
"DisplayApiDocumentation": false,
  "AllowedHosts": "*",
  "UpkeeperSettings": {
    "ConnectingClientsIsOnPremises": true,
    "ApplicationCheckInterval": 2,
    "MaxConcurrentClientChanges": 0,
    "UseHeaderAuthorization": true,
    "WebApiEndpoint": "https://[ADMIN_API_URL]/api",
    "WebApiAccessToken": "[SERVICEUSER_ACCESS_TOKEN]",
    "InstallationId": "3a33bd45-ce0b-4644-896b-ad1350ace001"
  }
}
```

Specification for settings

- **DisplayApiDocumentation** = Specify if API documentation should be accessible.
- **AllowedHosts** = Used to filter access. Default is (*) all clients allowed.
- **ConnectingClientsOnPremise** = Default value is false. If set to true, all clients connecting thru this ClientApi will be registered as on premise.
- **ApplicationCheckInterval** = Default value is 2. If lower data will be updated more frequently by will also load the database server harder. Higher value will make changes to clients less frequently but lower the load on the database server.
- **MaxConcurrentClientChanges** = Default value is 0 = no limits. This will limit the number of concurrent changes to specified number of clients. If limit is reached each new client connection will be refused. When changes are done concurrent number of clients will lower and new clients can connect. This feature lowers load on network and file shares.
- **UseHeaderAuthorization** = Default value is false. If set to true API will require client to connect with correct request header information. This feature requires client version 4.7 or higher.
- **WebApiEndpoint** = Set address to Admin API. Used for Log File function.
- **WebApiAccessToken** = Set access token from a user with Log File permissions.
- **InstallationId** = Default value is a Guid value. Value is used to identify each instance of a client API installation. This value must be changed in environment with multiple client API instances/installations.

Test – upKeeper Admin Web

Open a browser and write your address to upKeeper Admin Web. Result should look like this



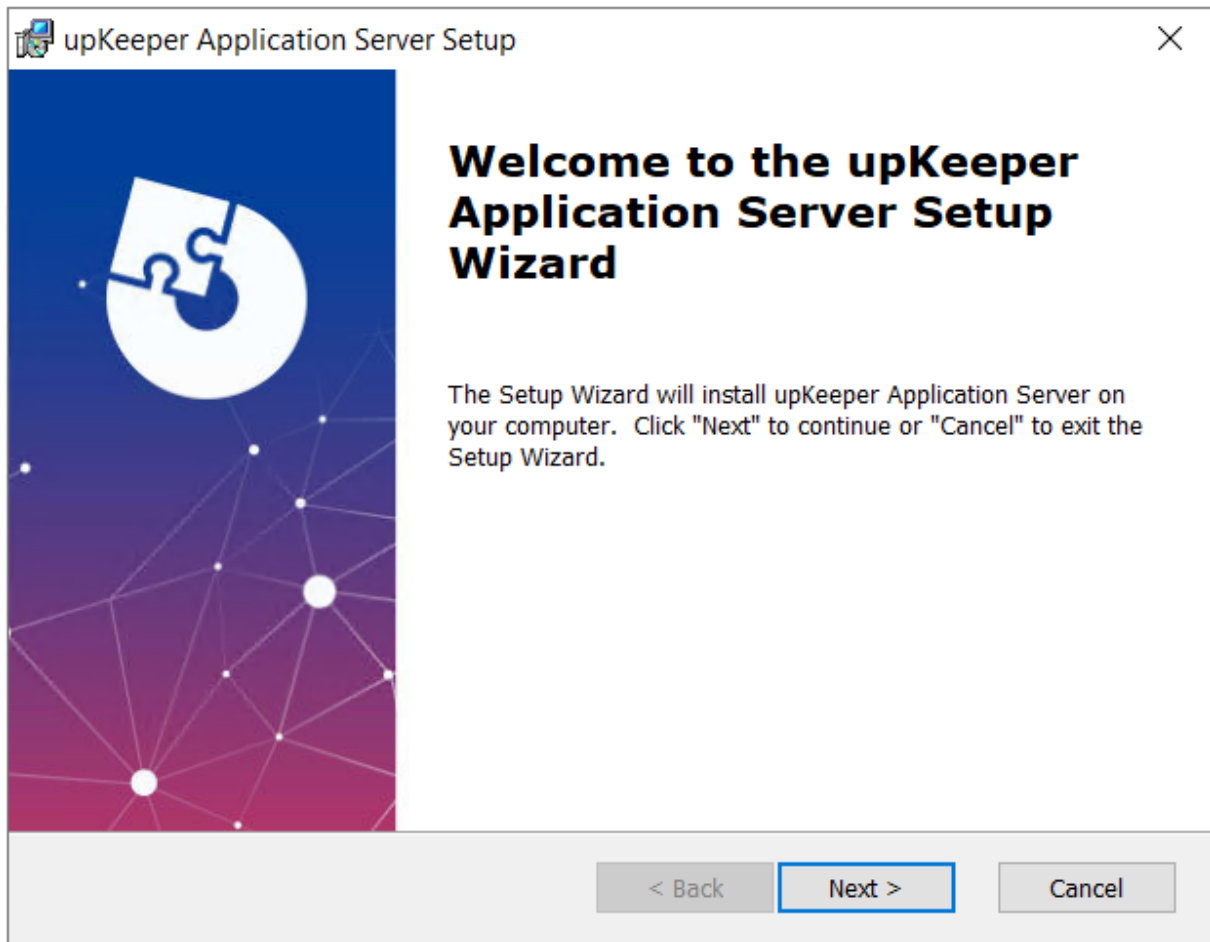
Troubleshooting

upKeeper Admin Web feels slow – check that upKeeper Admin Web and upKeeper Admin API write log files to specified log folders. If logs are missing, create necessary folder/folders and assign correct rights for user running the application pool for each website.

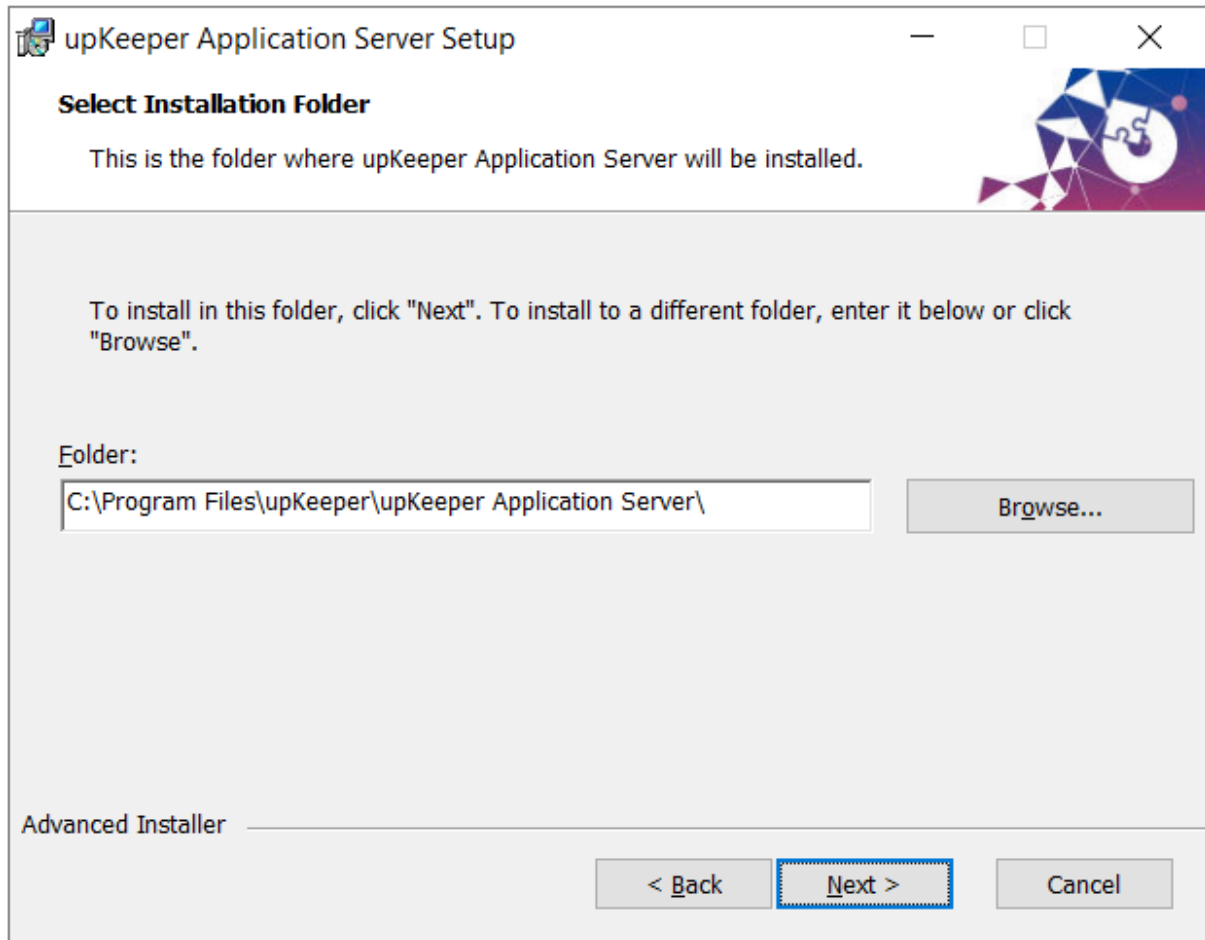
upKeeper Admin Web shows an error indicating URL address is too long – check that upKeeper Admin Web site user has write access to “log” and “dbkeys” folders.

Installation – upKeeper Application Server

Sign in with administrative rights to the server that will be used for the upKeeper Application Server.
Execute the file **upKeeper Application Server 6.X.X.X.msi**



Specify the **Destination Folder** for the installation and click Next



Fill out the form with information from database installation. Database server can be a network name or an ip address, include instance name if applicable. Click Next to continue.

The screenshot shows a window titled "upKeeper Application Server Setup" with a close button in the top right corner. The window is divided into two main sections. The top section is titled "Database Configuration" and contains the instruction "Enter database connection details". The bottom section contains four input fields: "Database server:" with the value "localhost", "Database name:" with the value "upkeeper", "User:" with the value "upkeeper_appuser", and "Password:" with a masked password of 12 dots. At the bottom of the window, there is a footer area labeled "Advanced Installer" and three buttons: "< Back", "Next >" (which is highlighted with a blue border), and "Cancel".

upKeeper Application Server Setup

Database Configuration

Enter database connection details

Database server:
localhost

Database name:
upkeeper

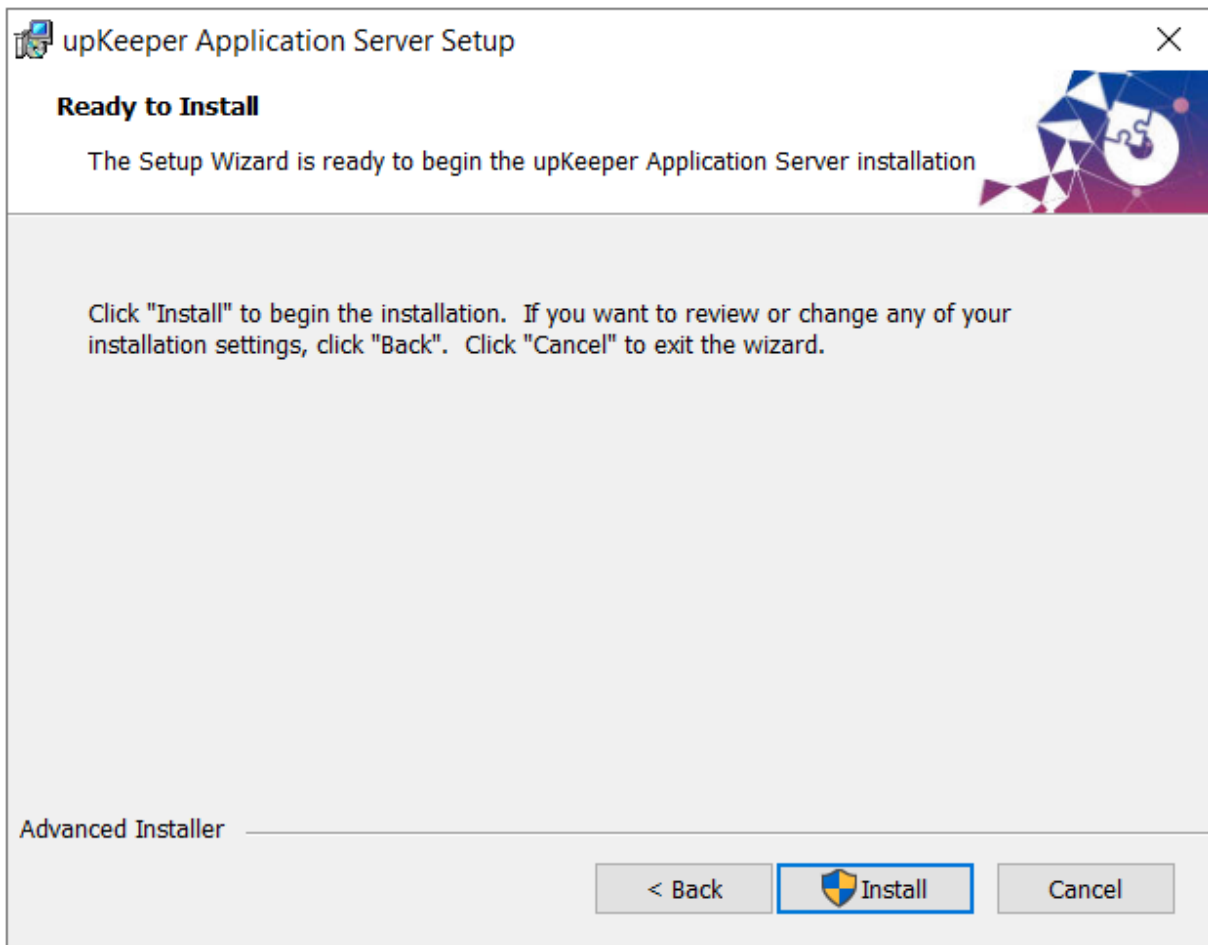
User:
upkeeper_appuser

Password:
●●●●●●●●●●●●

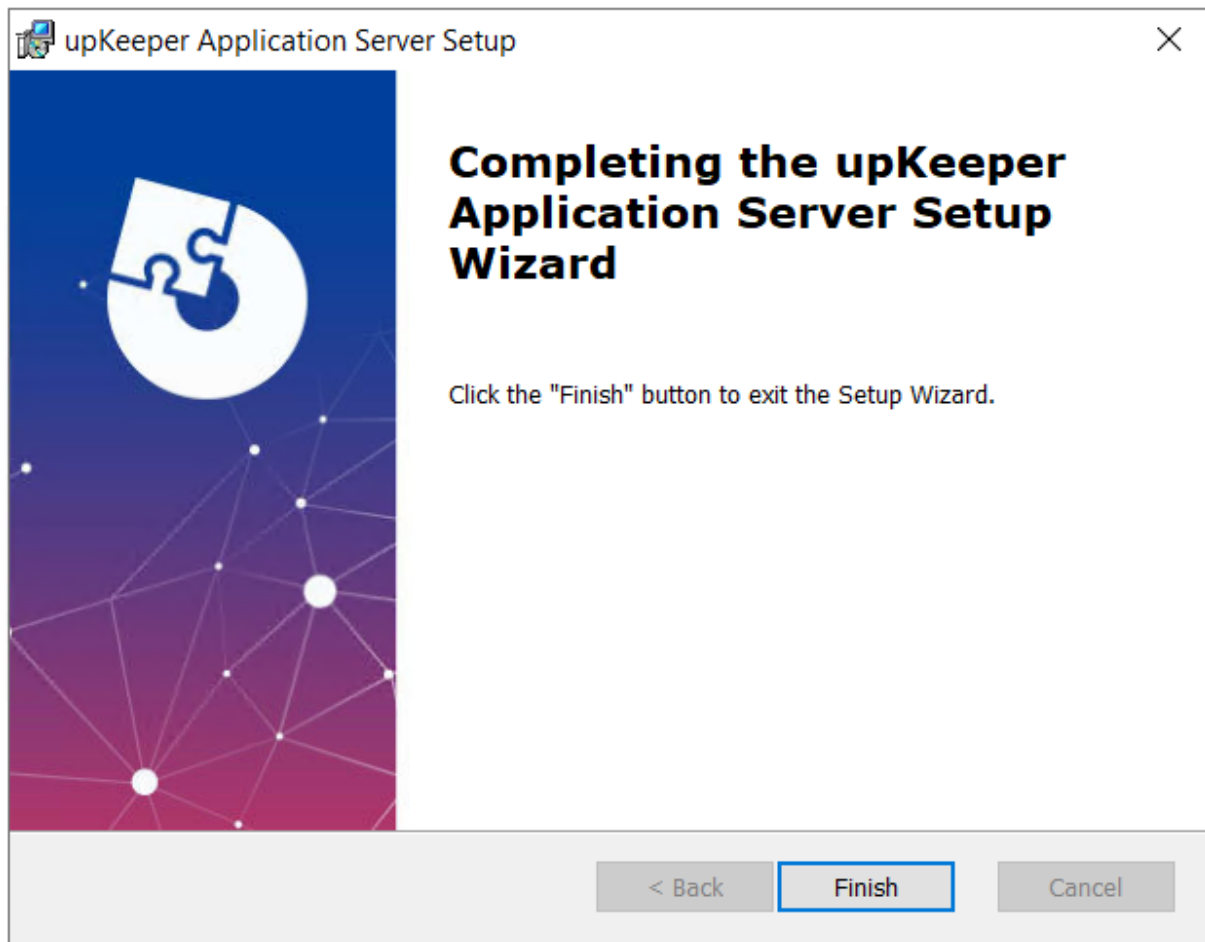
Advanced Installer

< Back Next > Cancel

Click Install to perform the installation.



Click Finish to close the installation dialog.



Edit the Application Server configuration file

Make the following changes to the file **appsettings.json**, located in the **Destination Folder** specified during the installation.

```
{
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "UpkeeperDb": " Data Source=[DATABASE_SERVER];Initial Catalog=[DATABASE];User
ID=[DATABASE_USER];Password=[DATABASE_PASSWORD];Integrated
Security=False;MultipleActiveResultSets=True;"
  },
  "UpkeeperSettings": {
    "IsMaster": true,
    "MasterSettings": {
      "WSUSBaseUrlApi": "http://192.168.90.19:9001",
      "WSUSNumberOfDownloadRetries": 12,
      "WSUSDelayOnRetry": 4000,
      "UpdateWorkDir": "C:\\updatework",

      "WarrantyApiEndPointDell": "https://apigtwb2c.us.dell.com/PROD/sbil/eapi/v5/asset-entitlements",
      "WarrantyApiTokenEndPointDell": "https://apigtwb2c.us.dell.com/auth/oauth/v2/token",
      "WarrantyApiManufacturerNamesDell": [
        "Dell",
```

```

        "Dell EMC",
        "Dell Inc."
    ],
    "WarrantyApiEndPointFujitsu": "https://aftersales.ts.fujitsu.com/wswcm/wswcm.asmx",
    "WarrantyApiManufacturerNamesFujitsu": [
        "Fujitsu",
        "Fujitsu Siemens"
    ],
    "WarrantyApiEndPointLenovo": "https://supportapi.lenovo.com/v2.5/warranty",
    "WarrantyApiManufacturerNamesLenovo": [
        "Lenovo",
        "IBM"
    ],
    "WarrantyApiRegexpLenovo": "^\\d{1,2}[a-zA-Z]{1}$"
},
"SyncAD": true,
"ThreadedHandlers": true,
"MasterService": "net.tcp://localhost:8889/Server",
"ExternalPort": 80,
"DistributionArea": "D:\\UpKeeperShare",
"StagingArea": "D:\\upKeeperStage",
"WSBPlatforms": [
    "Windows.Desktop"
],
"ApplicationCheckInterval": 2,
"MaxConcurrentClientChanges": 100,
"LicenseService": "http://license1.upkeeper.se/WebServices/Customers.svc",
"SaveLogMonths": 6,
"SaveComputerCountMonths": 5,
"SaveInventoriesPerComputer": 10,
"RemoveDeletedObjects": true,
"GetWSUSUpdatesFromUTC": "2022-02-23 00:18:01",
"AutopilotFactoryResetKeepEnrollmentData": false,
"AutopilotFactoryResetKeepUserData": false,
"DatabaseTimeout": 30
}
}

```

- **UpkeeperDb** = connection settings for upKeeper database. Replace values with brackets according to your environment.
- **IsMaster** = should normally be set to **true**.
If there are multiple Application Servers in the installation only one can be master, the others should be set to **false**.
- **WSUSBaseUrlApi** = address to the WSUS server where the upKeeper WSUS service is installed. Port is set to 9001 and cannot be changed.
- **WSUSNumberOfDownloadRetries** = max number of retries for a download
- **WSUSDelayOnRetry** = time between retries
- **UpdateWorkDir** = path where files to updates are downloaded and built into packages.
- **WarrantyApiEndPoint<manufacturer>** = Endpoint URLs to the manufacturer's warranty lookup services.
- **WarrantyApiTokenEndPoint<manufacturer>** = Endpoint URL to get manufacturer token.
- **WarrantyApiManufacturerNames<manufacturer>** = Manufacturer names to match a specific warranty service endpoint.
- **WarrantyApiRegexp<manufacturer>** = Custom filter expression to get correct warranty.
- **SyncAd** = Specifies that this application server should synchronize with active directory. This setting requires the application server to be configured as master.
- **ThreadedHandlers** = Specifies application server activities to be executed in separated threads and thereby not interfering with each other.
- **MasterService** = the address of the master Application Server

- **ExternalPort** = the port used for communication with clients outside the LAN.
- **DistributionArea** = the path to the folder where the application packages generated by the Application Server are copied.
- **StagingArea** = path to folder where packages original files are located.
- **WSBPlatforms** = Configure which platforms should be downloaded from Windows Store for Business.
- **ApplicationCheckInterval** = Specifies the interval between requests that the clients get commands for application changes. If the clients are assigned multiple applications to install the pause between the installations will be shorter. This setting should only be altered if the Application Server is experiencing performance problems. The lowest allowed value is 1 and default value if not set is 8.
- **MaxConcurrentClientChanges** = Specified the number of concurrent client that can get application change instructions. Default there is no limit of concurrent clients that get application change instructions.
- **LicenseService** = the address of the upKeeper license server, should not be changed.
- **SaveLogMonths** = Specifies the number of months the upkeeper eventlog events will be saved before they are deleted. Default value if not set is 36.
- **SaveComputerCountMonths** = Specifies the number of months the number of active computers will be saved in the local database. Default value if not set is 12.
- **SaveInventoriesPerComputer** = Specifies the number of inventories per computer that will be saved. Default value if not set is 10.
- **RemoveDeletedObjects** = Should objects marked for deletion be removed.
- **GetWSUSUpdatesFromUTC** = Time for last synchronisation with WSUS server.
- **AutopilotFactoryResetKeepEnrollmentData** = Controls whether or not the computer will remain enrolled in Intune after an Autopilot factory reset.
- **AutopilotFactoryResetKeepUserData** = Controls whether or not the user data will be preserved after an Autopilot factory reset.

NOTE! The upKeeper Application Server must be restarted after you have made changes to the configuration file:

Logging settings

Log path and level can be changed in **logging.json** file.

Multiple log settings can be specified for different purpose.

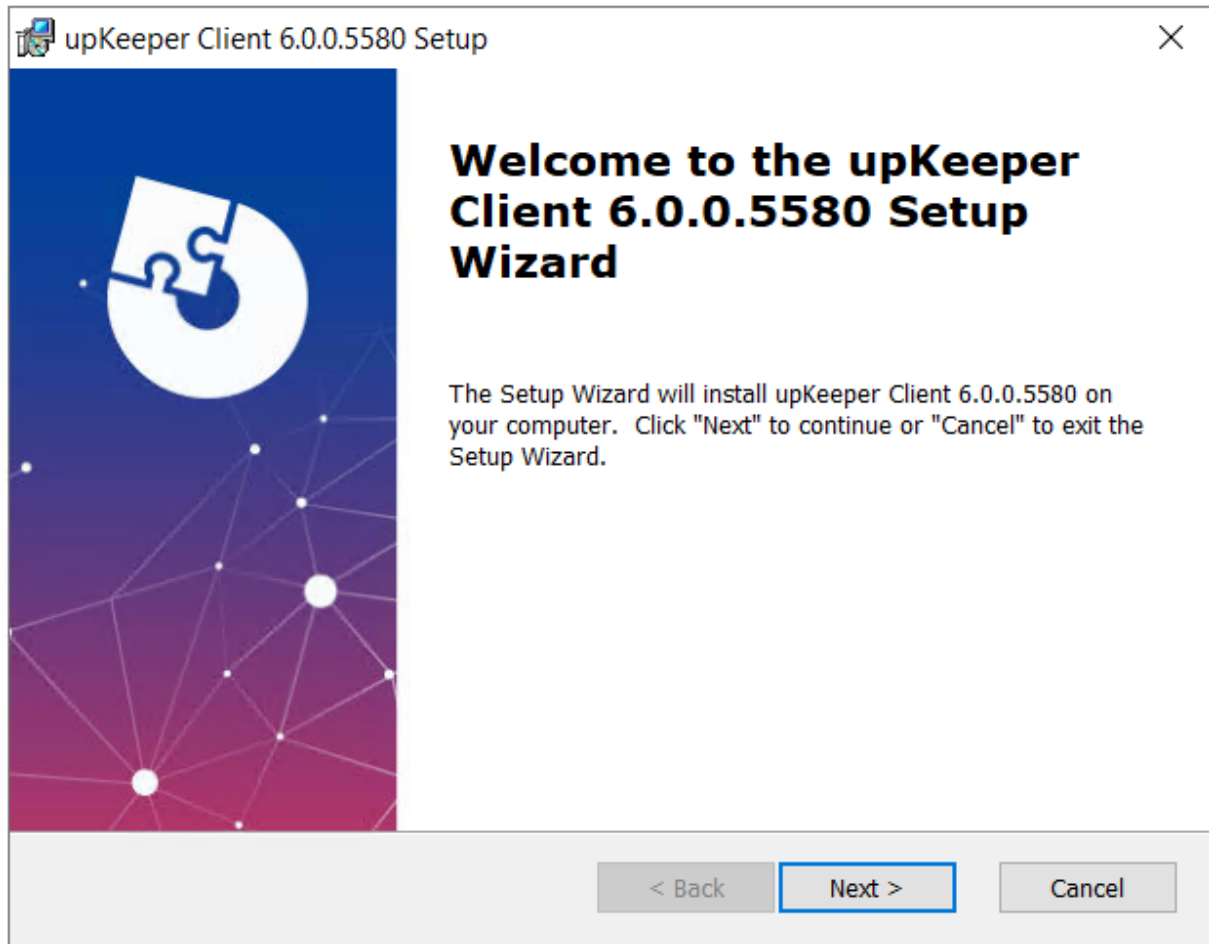
Log level is set in <level>. Approved values are ERROR, WARN, INFO and DEBUG. WARN is recommended for production environment, but INFO or DEBUG can be used in test- or troubleshot scenarios.

Optional service configuration for the upKeeper Application Server

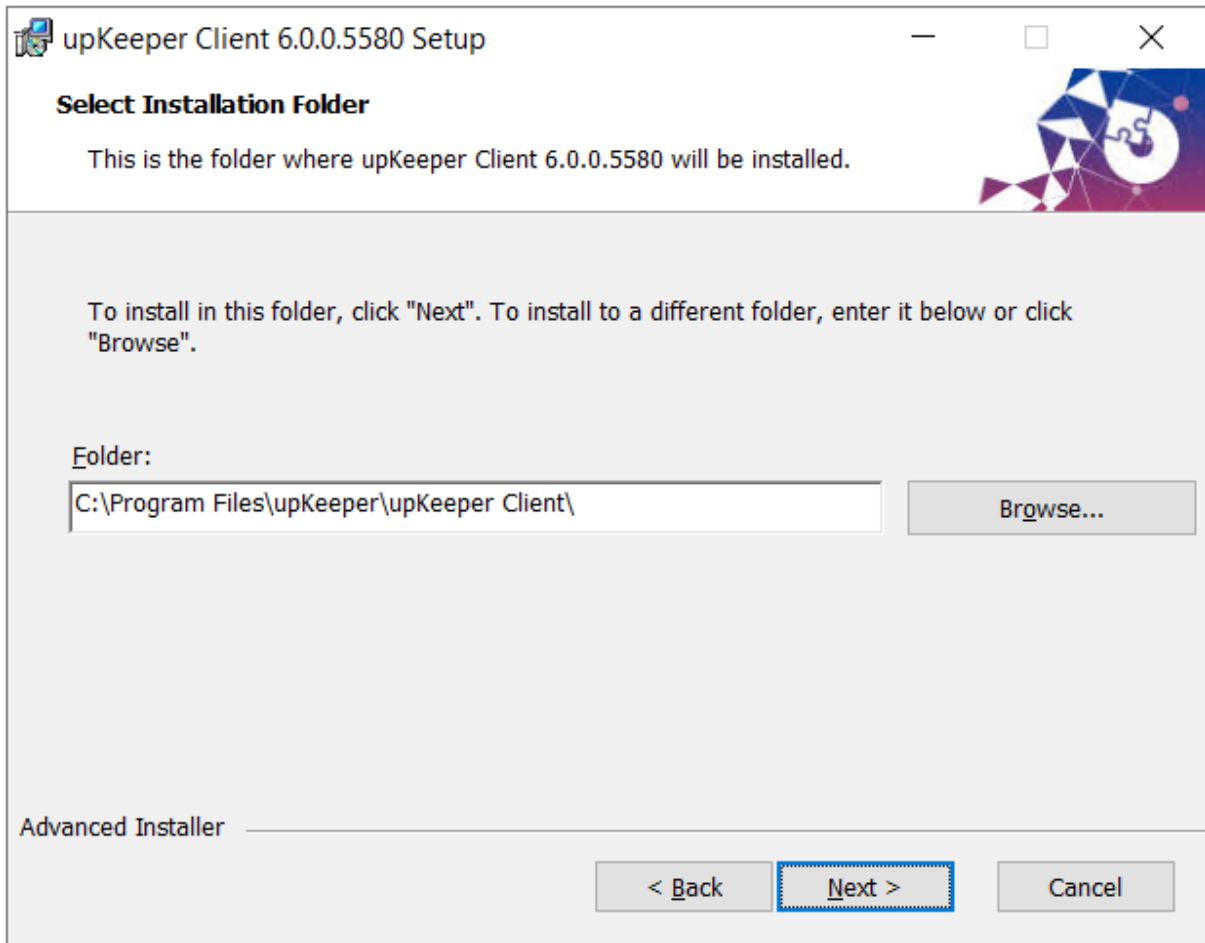
If the upKeeper Application Server is installed on another server than the file share, then the **upKeeper Application Server 6.X.X.X** service must run as a user account that has got the correct permissions in the domain.

Installation – upKeeper Client

Sign in with administrative rights to the client that will use upKeeper Client. Execute the file **upKeeper Client x64 6.0.0.5580.msi** or later. Click Next.



Specify the **Destination Folder** for the installation. Click Next.



The screenshot shows a Windows installer window titled "upKeeper Client 6.0.0.5580 Setup". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is titled "Select Installation Folder" and contains the following text: "This is the folder where upKeeper Client 6.0.0.5580 will be installed." Below this, there is a paragraph: "To install in this folder, click 'Next'. To install to a different folder, enter it below or click 'Browse'." A text input field labeled "Folder:" contains the path "C:\Program Files\upKeeper\upKeeper Client\". To the right of the input field is a "Browse..." button. At the bottom of the window, there are three buttons: "< Back", "Next >", and "Cancel". The "Next >" button is highlighted with a blue dotted border. The text "Advanced Installer" is visible in the bottom left corner of the window.

Enter **Organization Id** of the organization where the computer should belong. This is optional if your organization allow self-registration. Self-registration is not recommended in environments with multiple organizations. Click Next.

upKeeper Client 6.0.0.5580 Setup

Enter organization id

Organization id for the client.

The organization id for the client, a guid in the format:
11111111-2222-3333-4444-555555555555

Organization Id:
11112222-3333-4444-5555-666677778888

Advanced Installer

< Back Next > Cancel

Enter the endpoints at least **Endpoint1**, but two is preferred. This setting is optional and can be set with registry inputs at a later stage. Computer will start communicating as soon as endpoints are added, and service started. Click Next to continue.

upKeeper Client 6.0.0.5580 Setup

Enter endpoint URL

Endpoint that upKeeper Client 6.0.0.5580 will use.

The endpoint URLs. An example: https://server/

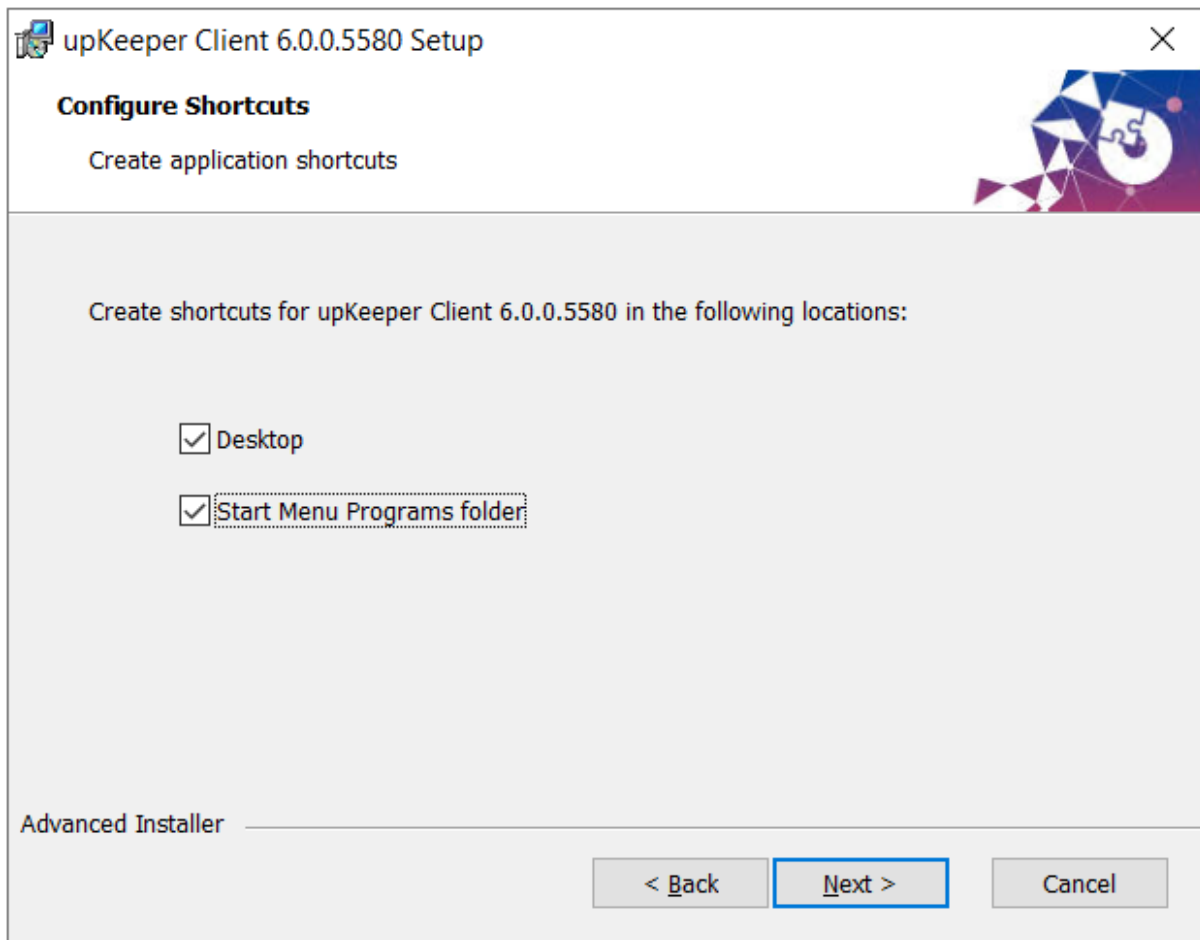
Endpoint1:
https://client1.upkeeperdev.local

Endpoint2:
https://client2.upkeeperdev.local

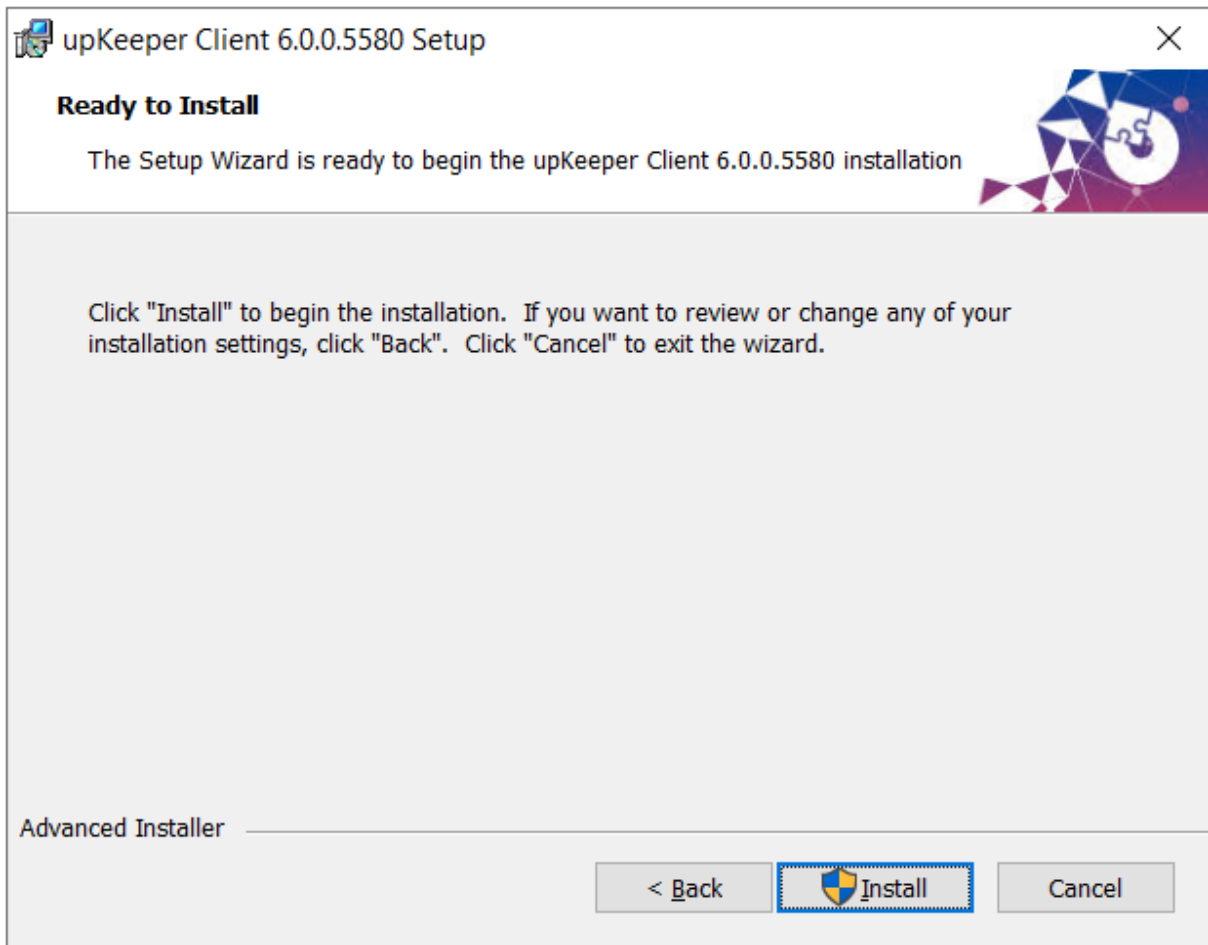
Advanced Installer

< Back Next > Cancel

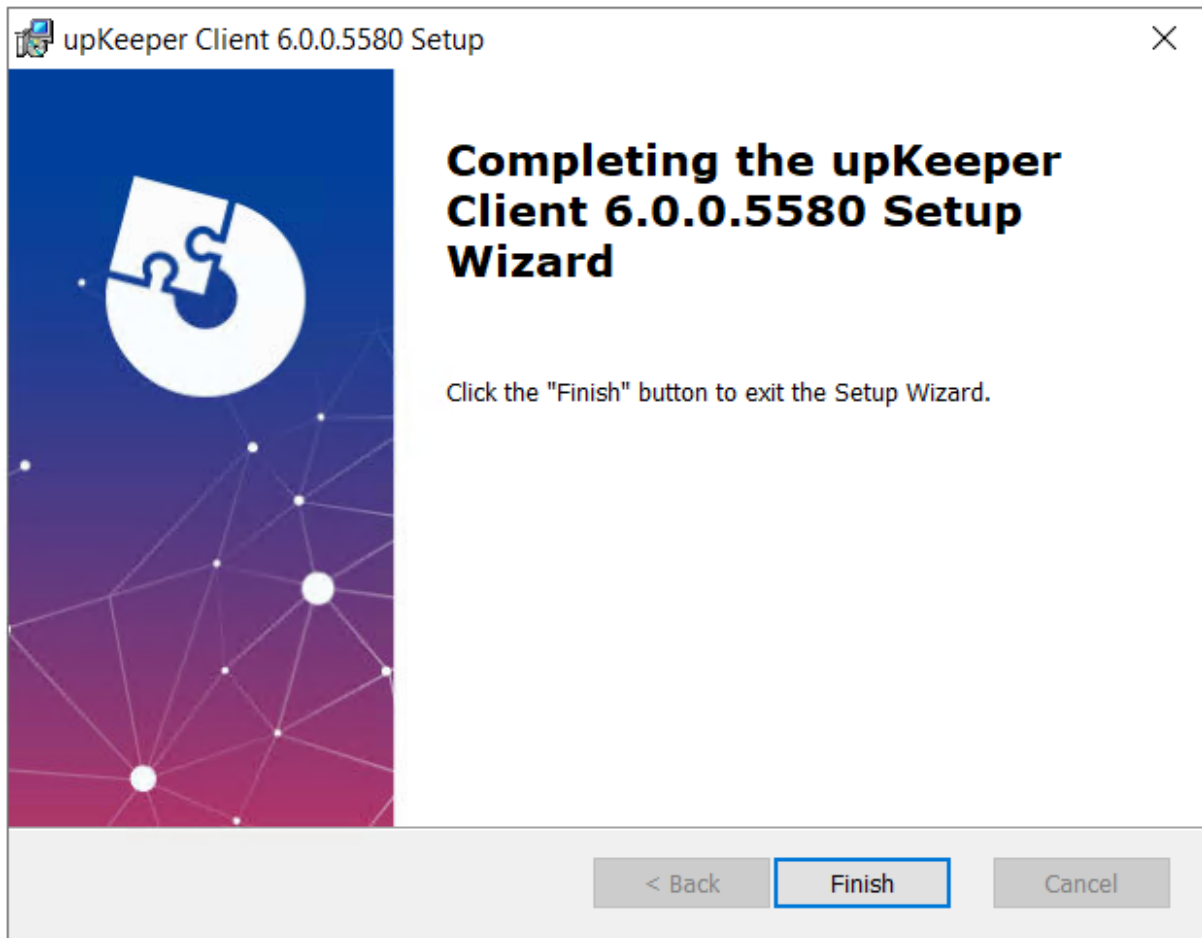
Specify if shortcuts should be created on desktop and/or start menu. Click Next to continue.



Click Install to perform installation.



Verify that installation is successful and click **Finish**.



Installation – upKeeper Client (silent)

Client can be installed silent setting all properties thru parameters.

upKeeper Client install parameters

- **INSTALLDIR** – Specify destination folder
- **ENDPOINT1**– Specify endpoint address used (ex ENDPOINT1=http://upkeeperclientapi/).
- **ENDPOINT2**– Specify endpoint address used (ex ENDPOINT2=http://upkeeperclientapi2/).
- **ORGANIZATIONID** – Specify which organization the client will self-register to. Only necessary if no computer object in upkeeper match the client’s properties.
- **MYUPKEEPERDESKTOPSHORTCUT** – Create shortcut to My upKeeper on desktop.
- **MYUPKEEPERSTARTMENUSHORTCUT** – Create shortcut My upKeeper in start menu.
- **NOSTART** – Client service will not start during installation.

Configuring – upKeeper Client

By default, all necessary client configuration is done during installation. Settings specified below are used in special occasions or when recommended by support personal. upKeeper Client connection interval is configured on the server side (minimum interval 10 seconds).

Note! Keys specified below must be entered in the following registry path:

HKLM\Software\upKeeper\Client

- **CleanUp** – Force client to clear application download area on client computer. Key is deleted after it has been used.
- **ResetComputerApplications** – Sets all applications assign to the computer to be installed. Key is deleted after it has been used.
- **IgnoreEnvironment** – Client will not check computer environment before installing or uninstalling applications. If upkeeper does not check the environment before making application changes the computer may pending for reboot, perform application or system changes which can result in conflicts or errors.
- **InventoryStartupTask** – upKeeper client perform a inventory on next startup. Key is removed when used.

Note! Key specified below must be entered in the following registry path:

HKLM\Software\upKeeper\Client\Settings

- **MailTo** – A supportbutton will be visible in the My upKeeper if this registry key exists and an email address is entered.
- **TakeSupportScreenshot** – If this registry key is entered and the value set to True/true, attach a screenshot of all screens in mail to support.
- **Popup** – If this registry key is entered and the value is set to disable, status popups will not be shown.
- **MyupKeeper** – If this registry key is entered and the value is set to disable, my upKeeper will not be accessible.
- **ColorScheme** – Changes the way alert messages form is colored. (0 = black/yellow, 1 = red/grey and 2 = flashing red/grey)
- **ApplicationTab** – If set to string value “disable” the application tab in My upKeeper will not be accessible.
- **OperatingSystemTab** – If set to string value “disable” the operating system tab in My upKeeper will not be accessible.

- **UserTab** – If set to string value “disable” the user tab in My upKeeper will not be accessible.
- **MeteringEnabled** – If set to true the client will report computer and application usage.
- **MeteringDirectories** – If other directories then program directories should be scanned for applications, set this value to directories of your choice, separated by semicolon (;).
- **MeasureFiles** – If specific files should be scanned, enter file names separated by colon (,).
- **Button1Text** – If text is entered a button with this text will be visible in My upKeeper.
- **Button1Command** – If text is entered it will be used as command if button with text from Button1Text is pressed/clicked.
- **Button2Text** – If text is entered a button with this text will be visible in My upKeeper.
- **Button2Command** – If text is entered it will be used as command if button with text from Button2Text is pressed/clicked.
- **HandlingAssignedApplications** – If set to string value “disable” assigned applications cannot be reinstalled.
- **Driveletter** – If set to specific letter client won’t assign drive letter when mapping SMB network shares. Normally client get first free driveletter starting from Z and goes up to A.
- **RequestApplication** – If set to string value “disable” the button for request applications will be disabled.
- **InventoryMissingUpdates** – If set to string value “disable” no inventory for missing updates will be performed.
- **MaximumMeasuringPoints** – By default upKeeper Client is measuring hardware resources every minute which is equivalent to 60 measure points per hour. You can increase or decrease the default value if need or recommended by upKeeper consultants. Clients are processing 1/4 of set measure points before reporting.
- **MinimumDiscSpaceRemainingPercent** – Defines when low free disc space information is sent. Default this value is set to 10 but can be changed to suite your needs.
- **MinimumMemoryRemainingPercent** – Defines when low remaining memory information is sent. Default this value is set to 10 but can be changed to suite your needs.
- **MaximumProcessorUsagePercent** – Defines when high CPU usage information is sent. Default value is set to 90 but can be changed to suite your needs.
- **SmbDelayAfterMountSeconds** – Defines delay time from when file share is mounted until client mount wim file (upKeeper Client 5.2.16 or above).
- **WimDelayAfterMountSeconds** – Default is one (1) second. Defines delay time from wim file is mounted until client starts to use application files (upKeeper Client 5.2.16 or above).

upKeeper Client Commands

upKeeper Client can run as an application to perform different tasks. Application file “upKeeper.Client.Manager.exe” can be found in the program folder where upKeeper Client is installed.

Available commands

Commands are added as parameters to exe-file. Structure: upKeeper.Client.Manager.exe [command]

directTask – Command used to execute functions immediately. Functions available “inventory”.
Structure: directTask=inventory

Branding - My upKeeper Client

My upKeeper Client can be branded with custom title. To change form title, popup title and menu option you can add the registry string BrandingTitle with the value you wish to be shown. Registry string must be placed under the Settings key. (HKLM\Software\upKeeper\Client\Settings)

upKeeper Client Logger

upKeeper Client Logger is a separate component installed with the upKeeper Manager Client and can be found in the same installation path as the other client components. upKeeper Client Logger is used for custom logging in scripts or commands to send information and files that will be shown in the upKeeper Manager Admin Web. Messages or files sent using this function will be found in corresponding device name under the “Eventlog” tab and files will be found under “Advanced” tab.

Usage: upKeeper.Client.Logger.exe [<command>]

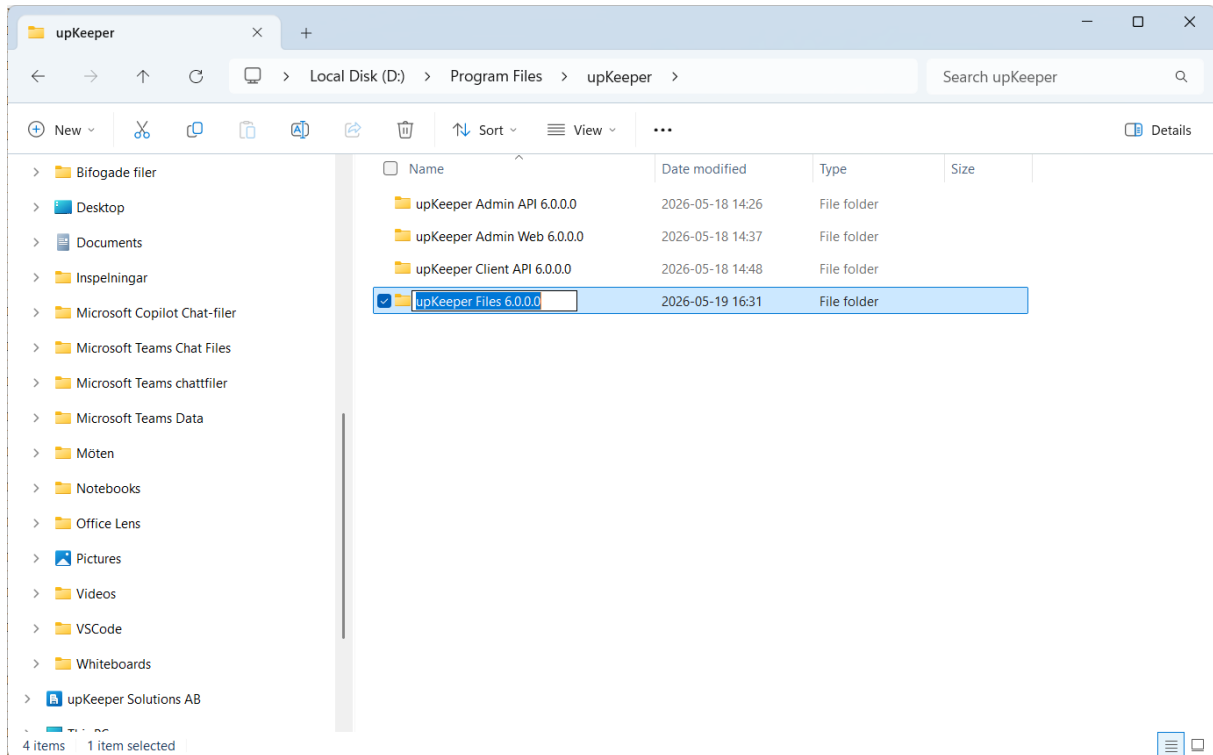
Commands available is dependant on version and can be found executing exe file without commands or with parameter -h or –help.

Installation - upKeeper Files Website

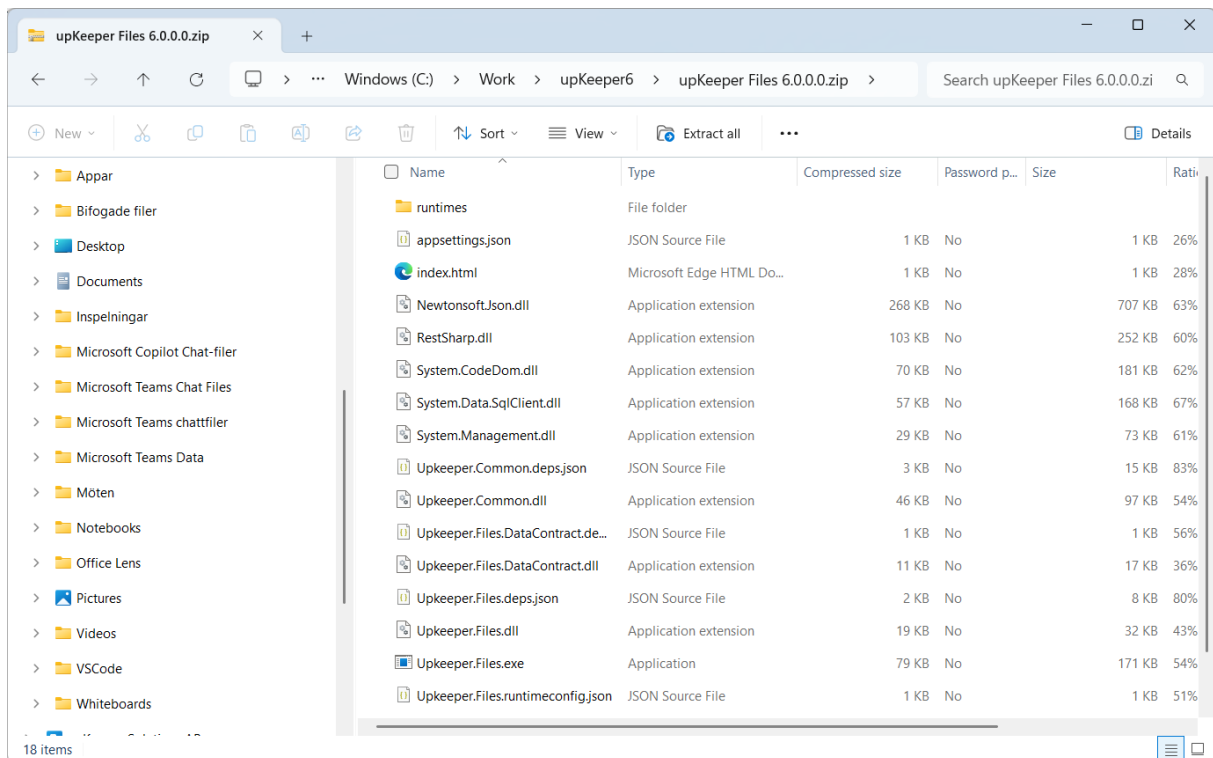
The upKeeper Files Website handles distribution of application packages over HTTP. It must be installed if you want to manage clients not located on the LAN.

Sign in with administrative rights to the server you used for the upKeeper Files.

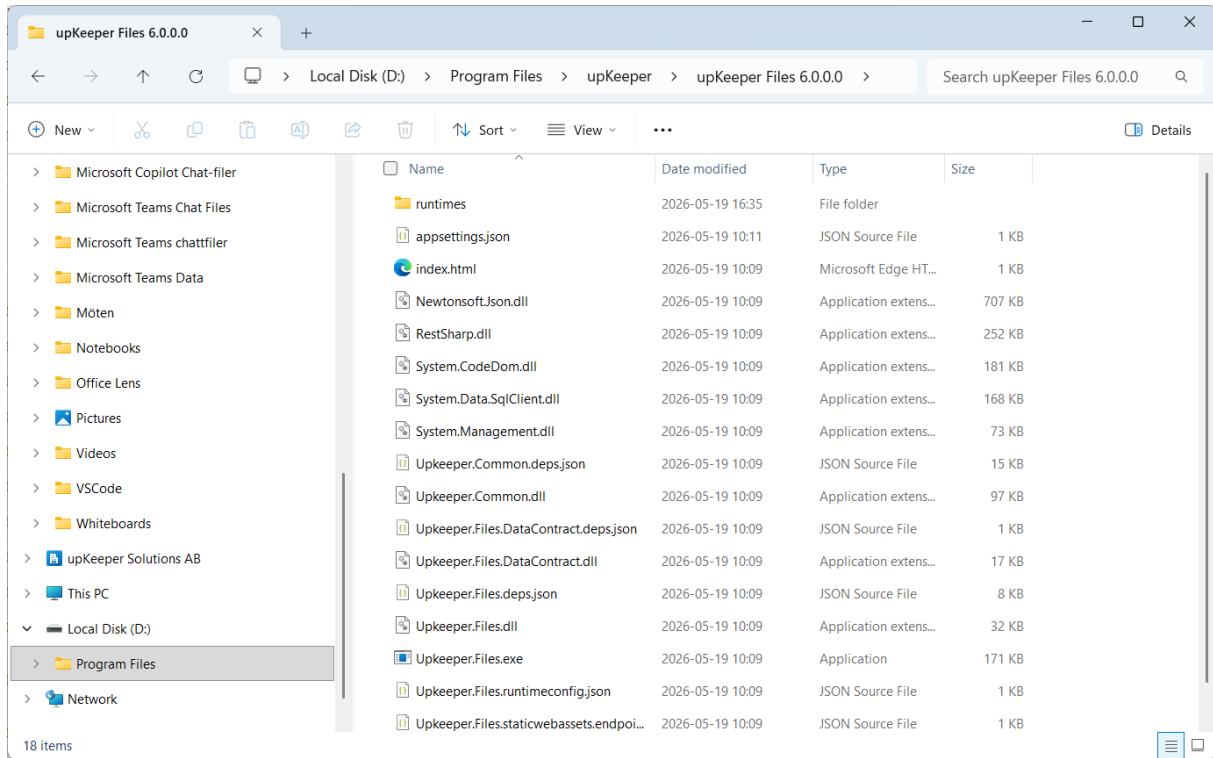
Create a folder named according to current upKeeper Files zip-file in the path “C:\Program Files\upKeeper”.



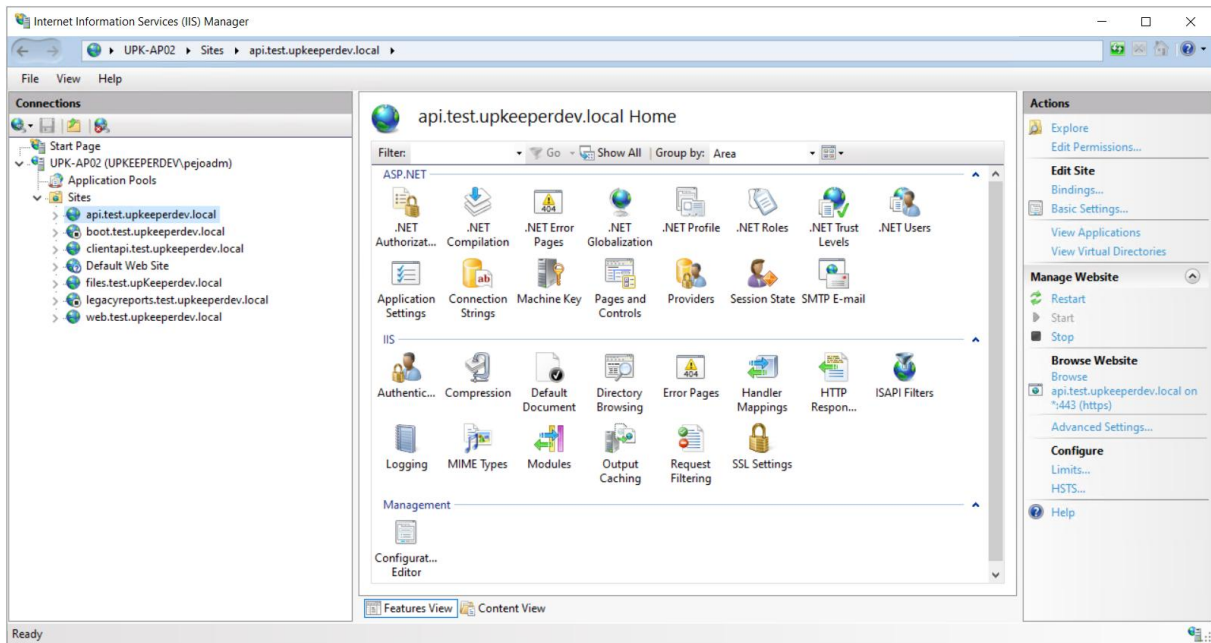
Extract the content of the file “upKeeper Files 6.x.x.zip” into the folder just created.



The upKeeper Files folder after extraction. (Content can look different due to version)



Open “Internet Information Services (IIS) Manager” on the server



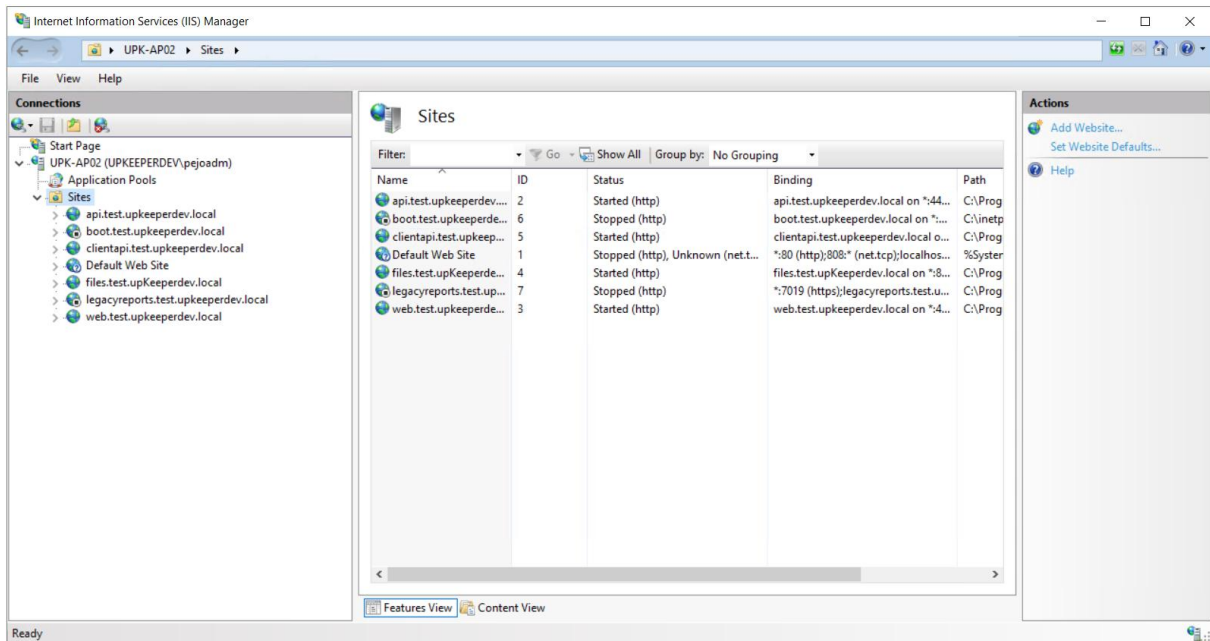
Add new web site in “Internet Information Services (IIS) Manager”. The “Physical path” should point to the folder where you extracted the web files. Remember to change the “host name” to a DNS address reachable from the clients.

The screenshot shows the "Add Website" dialog box in IIS Manager. The fields are filled as follows:

- Site name:** files.test.upkeeperdev.local
- Application pool:** files.test.upkeeperdev.local
- Content Directory:**
 - Physical path:** C:\Program Files\upkeeper\upKeeper Files 6.0.0.0
 - Pass-through authentication:** Connect as... Test Settings...
- Binding:**
 - Type:** https
 - IP address:** All Unassigned
 - Port:** 443
 - Host name:** files.test.upkeeperdev.local
 - Require Server Name Indication
 - Disable TLS 1.3 over TCP
 - Disable Legacy TLS
 - Disable OCSP Stapling
 - Disable QUIC
 - Disable HTTP/2
- SSL certificate:** *.test.upkeeperdev.local
- Start Website immediately

Buttons: OK, Cancel

Available web sites should look like this when all components is installed on one server. This server has upKeeper Admin API, upKeeper Admin Web, upKeeper Client API, upKeeper Files, upKeeper Legacy Reports and upKeeper PXE over HTTP boot (testing).



Edit the upKeeper Files Website configuration file

Make the following changes to the **upKeeper Files** configuration file: **appsettings.json**.

Edit the database connection settings according settings in upKeeper Admin API or upKeeper Client API. Content root path should be configured to the folder where all organization files, drivers and platform files are placed. Notice that you need to use double backslashes (\\) in path.

```
{  
  "ConnectionStrings": {  
    "UpkeeperDb": "Data Source=[DATABASE_SERVER];Initial Catalog=[DATABASE];User  
ID=[DATABASE_USERNAME];Password=[DATABASE_PASSWORD];Integrated  
Security=False;MultipleActiveResultSets=True;TrustServerCertificate=True"  
  },  
  "ContentRootPath": "c:\\temp\\"  
}
```

Test the configuration

You can test the settings by creating a text file in the folder where you installed upKeeper Files.

Then test to surf to your text file, **http://sitename/fil.txt**.

You will get a login window asking for a username and a password.

Because the website is intended to be used by the upKeeper client, the required credentials are the **Computer Id** and **Organization Id** of a managed client computer.

You will find those in the registry of a installed client: HKLM\SOFTWARE\upKeeper\Client\Settings

ComputerID = login name

OrganizationId = password

If everything is correctly configured, you will see the content of your text file in the browser.

Build upKeeper DSOS or SOS

To build or update upKeeper DSOS or SOS we have to create a build structure including upKeeper SOS files and software from Microsoft.

Install Microsoft software

Download and install “Windows ADK” and “Windows PE add-on for the ADK” on a computer where you want to maintain your upKeeper SOS files.


1. Download “Windows ADK” for Windows 10 or later and “Windows PE add-on for the ADK” from Microsoft website.
2. Run “adksetup.exe” and choose “Deployment Tools” and “Configure Designer” and install in default location. Other components can be added.

Select the features you want to change

Click a feature name for more information.

- Application Compatibility Tools
- Deployment Tools
- Imaging And Configuration Designer (ICD)
- Configuration Designer
- User State Migration Tool (USMT)
- Volume Activation Management Tool (VAMT)
- Windows Performance Toolkit

3. Run “adkwinesetup.exe” and install with default settings.

 Windows Assessment and Deployment Kit Windows Preinstallation Environment Add-ons - Windows 10

Select the features you want to change

Click a feature name for more information.

- Windows Preinstallation Environment (Windows PE)

Windows Preinstallation Environment (Windows PE)

Minimal operating system designed to prepare a computer for installation and servicing of Windows.

Includes:

- Windows PE (x86)
- Windows PE (AMD64)
- Windows PE (ARM)
- Windows PE (ARM64)

Add upKeeper DSOS or SOS files

Download latest version of upKeeper DSOS or SOS files and make necessary and optional configuration.

1. Download “update_sos.zip” for latest version of upKeeper Manager and extract to “C:\upKeeper\upKeeperSOS” on your computer.

- Download “upKeeper SOS x.xx” for latest version of upKeeper Manager and extract to “C:\upKeeper\upKeeperSOS\upKeeper” on your computer.

Configure upKeeper SOS

Open C:\upKeeper\upKeeperSOS\upkeeper\Upkeeper.Sos.exe.config in a text editor and edit to match configuration for your upKeeper Manager installation.

File Edit Format View Help

```
<appSettings>
  <add key="Id" value="upKeeperSOS_w10_amd64" />
  <add key="ShowComputerInformation" value="True"/>
  <add key="Endpoints" value="https://upkeeperClientapi.upkeeper.se" />
  <!--<add key="Background" value ="white" /> |-->
  <!--<add key="IgnoreSMBFailles" value="true" /> -->
  <!--<add key="passphrase" value="Roligt" /> -->
</appSettings>
```

Settings

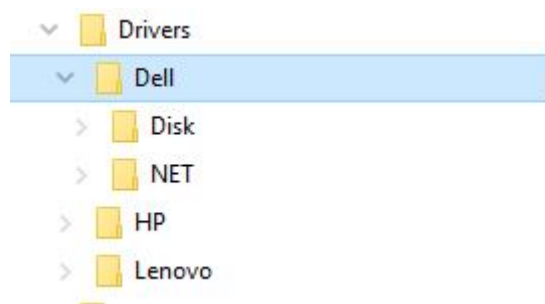
- **Id** – Set to folder name of your upKeeper SOS files on distribution point.
- **ShowComputerInformation** – If true SOS will show detailed computer information.
- **Endpoints** – address/addresses to client API endpoints of your installation. Multiple endpoint addresses are separated by comma (“,”) without any space.

Optional settings

- **Background** – set to “white” will change background to white and text to black.
- **Passphrase** – will connect to organization with specified passphrase if computer is not recognized.
- **IgnoreSMBFailles** – set to “True” will force SOS to proceed even if download of files from SMB distribution point fails.

Add drivers to upKeeper SOS

upKeeper SOS needs in some case unique drivers for network and/or disc to be able to run. Download drivers that are built for the version of WinPE you are using. Add drivers to a folder structure that are easy to read and update (see example below).



Update upKeeper DSOS or SOS

After performing the steps above you are ready to update upKeeper SOS.

1. Run "update_sos.cmd" in "Run as administrator" mode and verify that every step are successful.
2. Copy folder "upKeeperSOS_w10_amd64" to upKeeper distribution point(s) and add the new "boot.wim" file to all USB drives and WDS server.
3. Boot client on USB drive or WDS to verify new SOS.

Known issues

Files that have been download are blocked, start powershell, go to upKeeperSOS\upkeeper and run

```
"dir | Unblock-File"
```

Run update upKeeper SOS.

Tips!

You can use WDSUTIL (See example command below) to add or update upKeeper SOS to WDS, remember to change path to image file.

```
WDSUTIL /Replace-Image /Image:"upKeeperSOS" /ImageType:Boot /Architecture:x64  
/ReplacementImage /ImageFile:"\\upKeeperDIST\upKeeperSOS_w10_amd64\Sources\boot.wim"  
/Name:"upKeeperSOS" /Description:"upKeeperSOS"
```

Diagnosics in upKeeper DSOS

upKeeper DSOS is default configured without diagnostics (formally known as debug) mode. You can add diagnostics mode by creating a password file using "Upkeeper.DSOS.exe" file and add the created file to the upKeeper DSOS file structure before creating or updating upKeeper DSOS.

Command: Upkeeper.DSOS.exe -Password ThisIsMyPassword

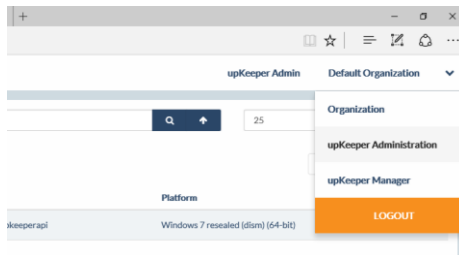
Configuration - upKeeper 5.X

The upKeeper Administration Website <http://upkeeperweb/> is where the upKeeper installation is managed.

Log on with the administrative user: **upkeeper** password: **upkeeper**.

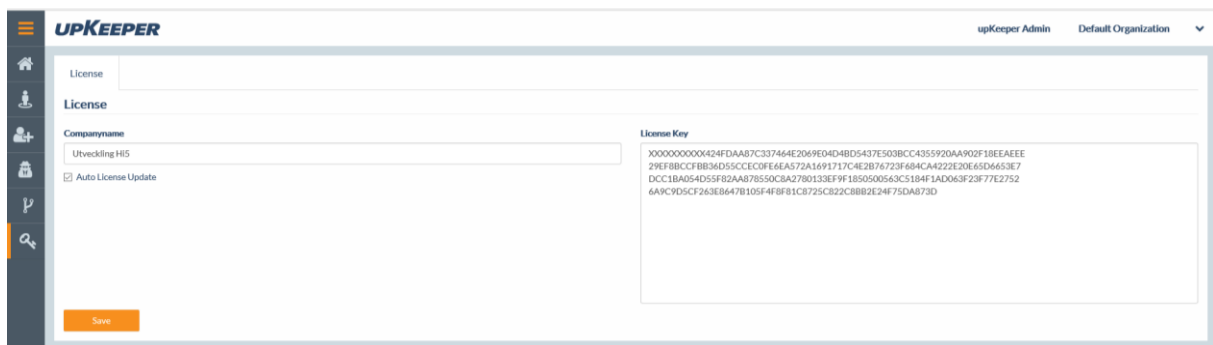
Configuration - Licence

Select **upKeeper Administration**



Open the tab **License**

Enter the name of the company holding the license for the installation under **Company Name** and click **Save**.



Normally the license key will be updated by the upKeeper Application Server within five minutes

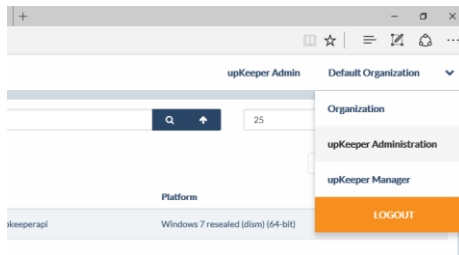
If you have been given a license key, you can enter it in the **LicenseKey** field.

Create a new Organization

An upKeeper installation can be divided in **Organizations**. This can be used to separate users and computers from different companies or departments to simplify management or delegate permissions.

On installation one Organization named **Default Organization** is created, this Organization can be renamed to something more descriptive.

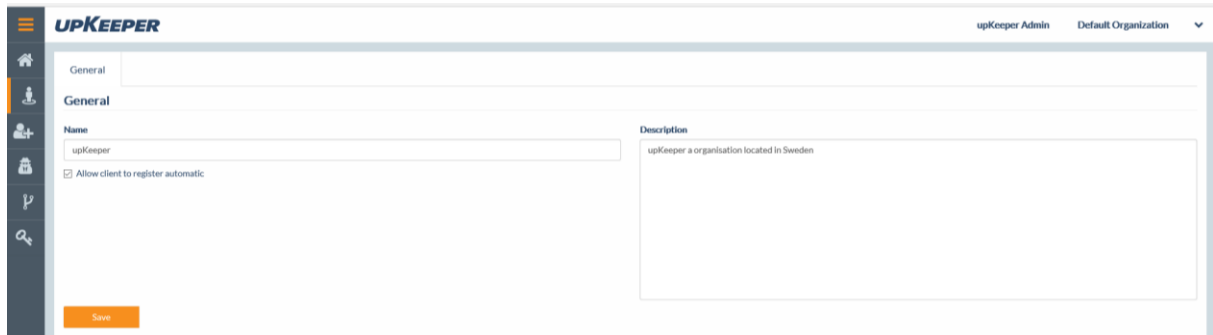
Select upKeeper Administration



Open the tab **Organizations** and select **Create**

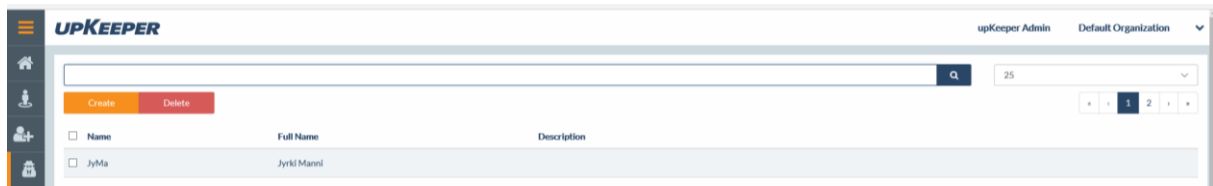


Enter a name for the Organization and check the box **Allow clients ...** if you want client computers to automatically be registered.

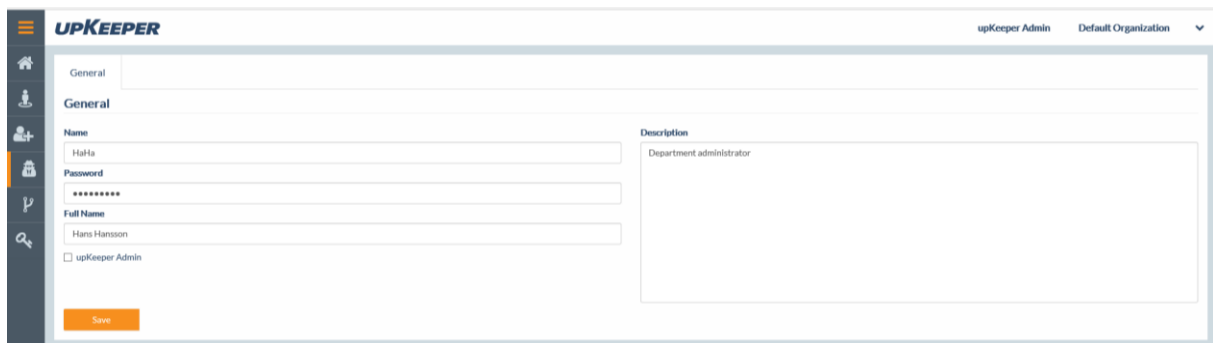


Create a User

Open the tab **Users** and select **Create**.

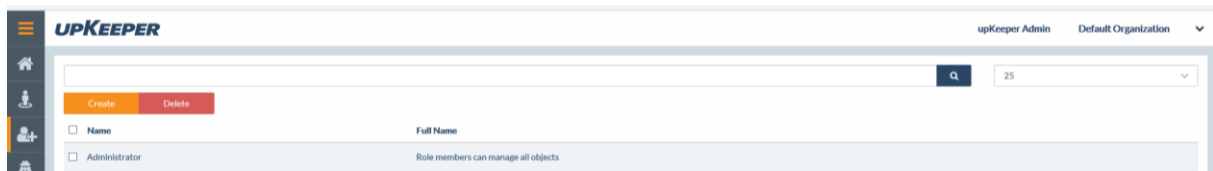


Enter the user name, password and the full name

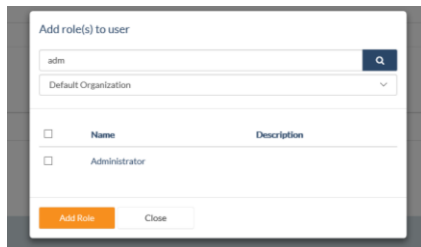


Do not check **upKeeper Admin** unless you want the user to have access to all Organizations.

Open the tab **Roles** and select **Add**

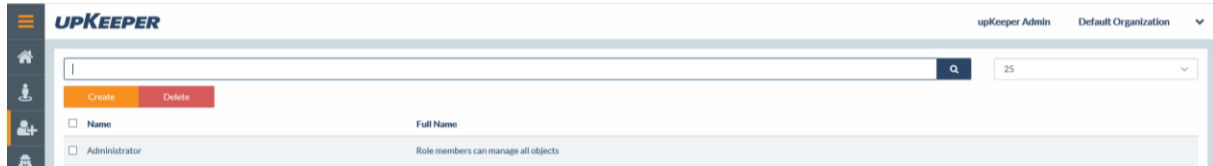


There exists only two roles by default, **Administrator** and **HelpDesk**, make sure that the right Organization is selected. Then select the role you want the user to have and click on **Choose Roles**. Then close the **Add roles to user** dialog.

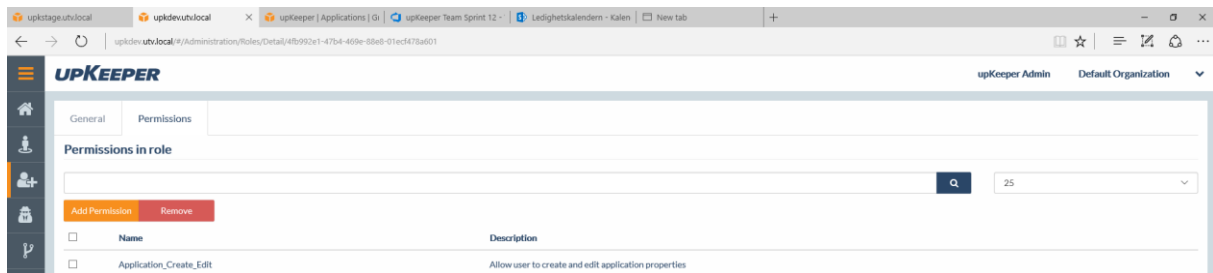


Create additional roles

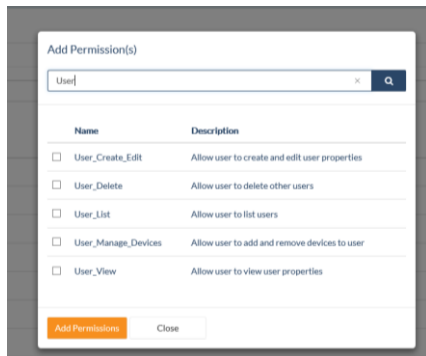
Open the tab **Roles** and click the **Create** button. Enter the name of the role you want to create (think of what the role will be used for e.g., Deploy Computers, Handle Applications, etc.)



Open the tab **Permissions** and click the **Add** button



Select the permissions that your role should have and click on **Choose permissions**, then close the **Add permissions to role** dialog.



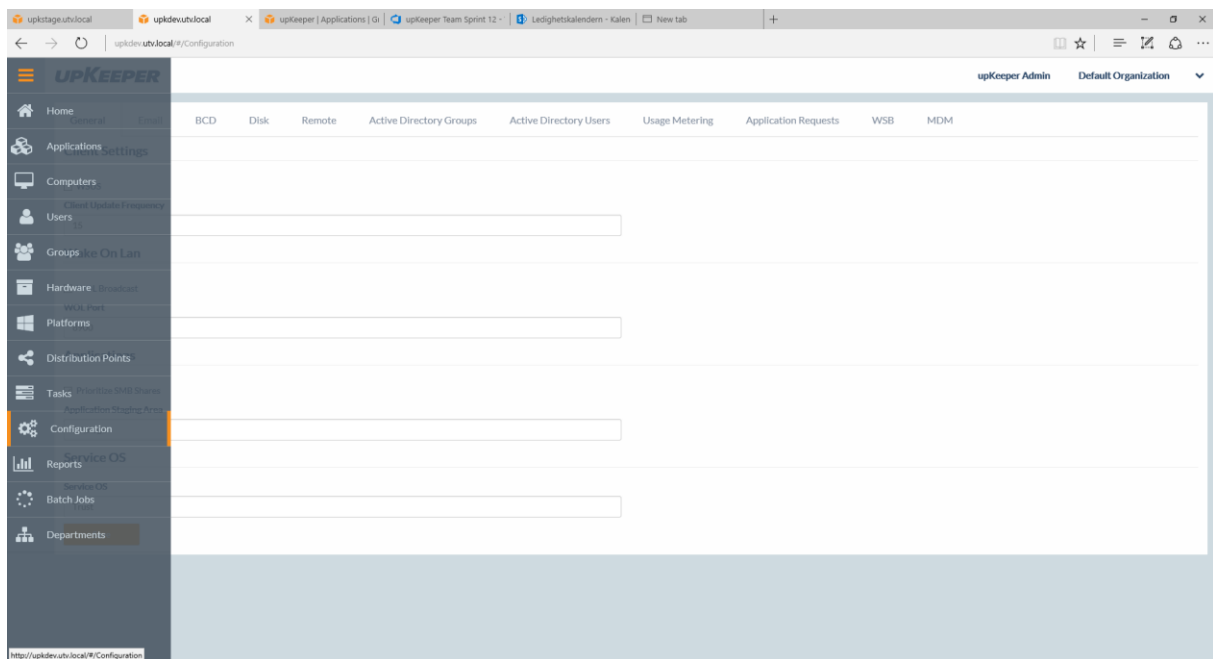
Configuration – Organization settings

If necessary, click the link **upKeeper Manager** in the upper right corner to leave the **upKeeper Administration** area.

Note! The settings in this section are made for each Organization.

Configuration – General

Start by open the **Configuration** tab.



The section **General** will be visible.

WSUS = Configuring the clients to communicate with the WSUS server in the final stages of the OS installation.

WOL Broadcast = Wake On Lan requests are made with both Unicast and Broadcast.

WOL Port = the Port for Wake On Lan requests.

Client Update Frequency = The interval in seconds between client computers connection with the server, default 15 sec.

Application Staging Area = The Path where installation files for the applications are saved. (The upKeeper Application Server then repackages the applications as .wim files that are distributed to the clients.)

Configuration – BCD

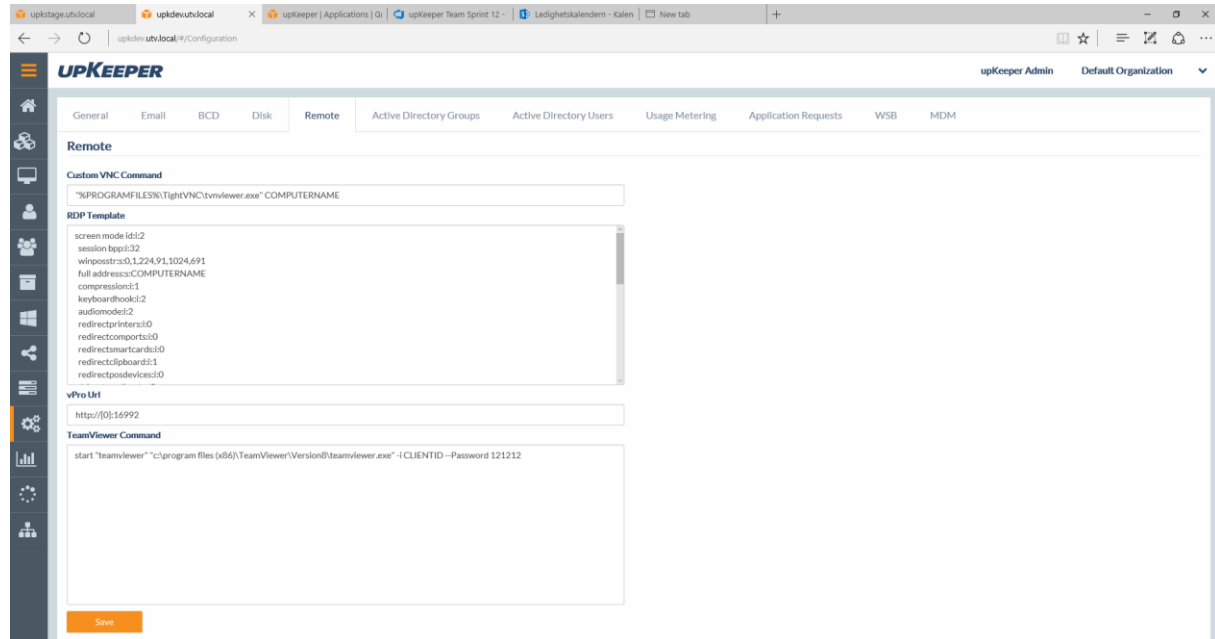
These settings should not normally be edited.

Configuration – Disk

Service Partition Size = The size of the service partition in megabytes.

Diskpart Install, Diskpart Reinstall and Diskpart Drive Letters = Determines how the client disk are partitioned and how drive letters are assigned.

Configuration – Remote



Custom VNC Command = The command to remotely control clients.

RDP Template = Template for Remote Desktop connections to clients.

The variable **COMPUTERNAME** will be replaced by the computer name and **CLIENTID** will be replaced by TeamViewer Id when the commands are executed.

vPRO URL = Address to access computers with vPro activated. {0} will be replaced with computer name.

TeamViewer Command = Command to start Teamviewer on the administrator PC and used collected and predefined information to connect directly to the client. CLIENTID or COMPUTERNAME will be replaced with proper values when executed. (Example: start "teamviewer" "c:\program files (x86)\TeamViewer\Version8\teamviewer.exe" -i CLIENTID --Password XXXXXX)

Configuration – Active Directory Groups

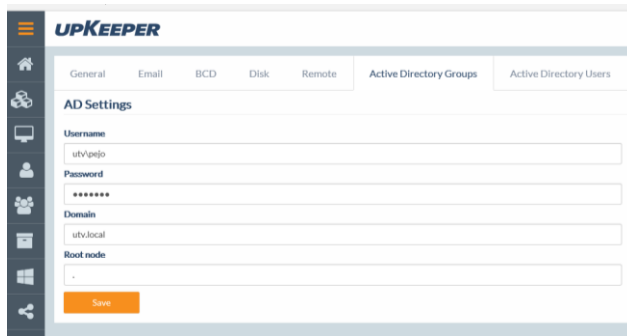
Configure synchronization of groups and computers from AD to upKeeper.

Specify the necessary information under **AD Settings** and click **Save**.

When the groups from AD has been read by upKeeper you can select the groups you want to synchronize by clicking **Add**.

Configuration – Active Directory Users

Configure synchronization of Users from AD to upKeeper. Specify the necessary information under **AD Settings** and click **Save**.



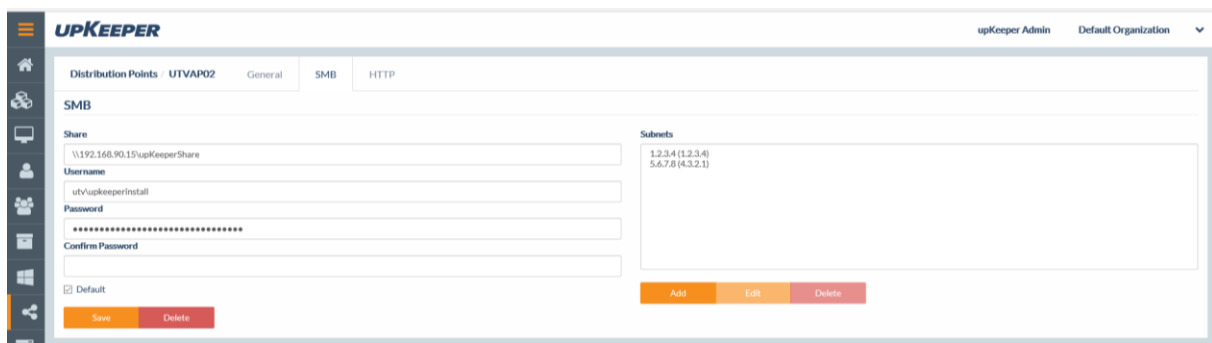
Configuration – Distribution Points

Open the tab **Distribution Points** and click on **Create**.



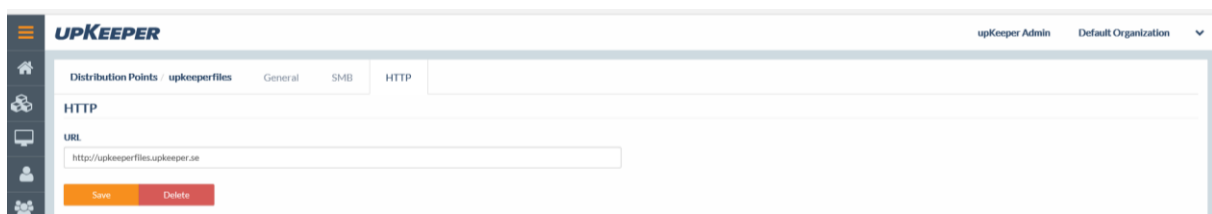
Enter **Name**, **Location** and **Description** and check **Active**, then click **Save**.

Open the **SMB** tab.



Enter the address of your file share, specify which account you want to use, enter the subnets involved and select if it should be the default Distribution Point. Then click **Save**.

Open the **HTTP** tab.



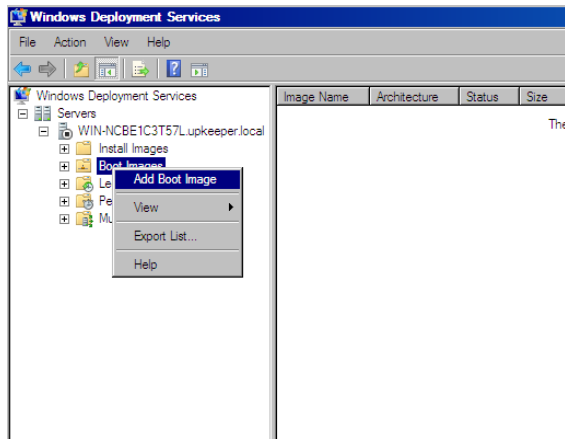
This setting is used to deploy applications to computers that are included in the upKeeper database and have a connection with upKeeper over the Internet.

Enter the web address for the site used by upKeeper to distribute applications. Remember that you have to configure your DNS so that the clients can find it.

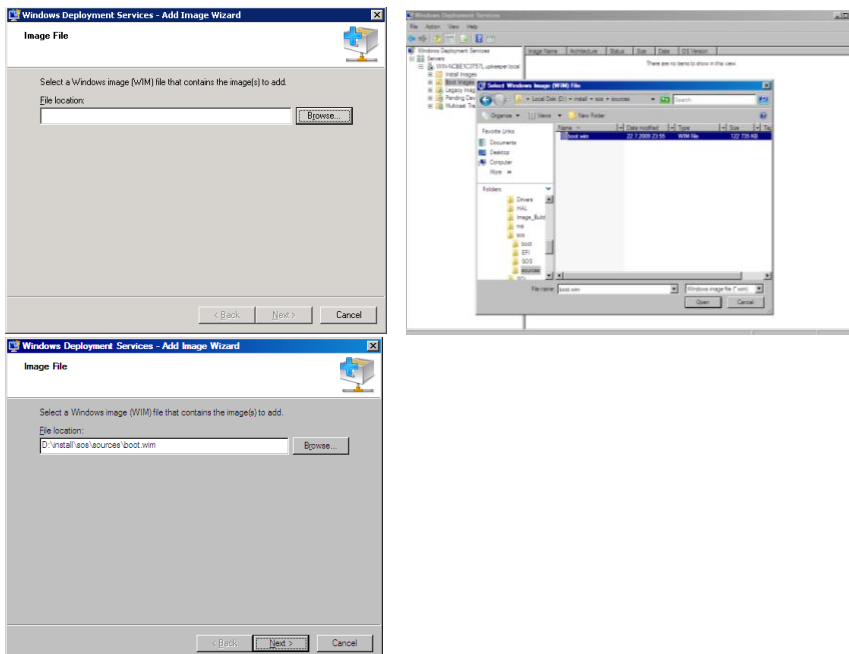
Appendix A - Configuration of Windows Deployment Services

- Sign in with administrative rights to the server that will be used for Windows Deployment Services.
- Start Windows Deployment Services.

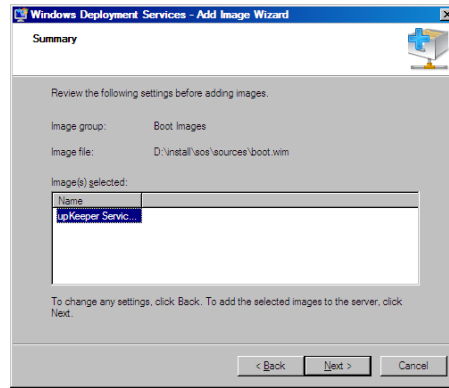
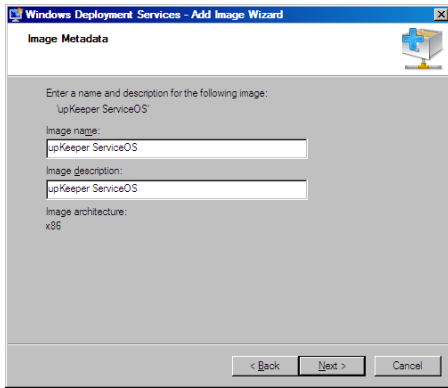
Expand the tree on the left side, find the current server and right click on **Boot Images**.
Select **Add Boot Image**.



Click **Browse** and browse to the folder where your upKeeper ServiceOS files are located.
Select the file **Boot.wim**.



Enter an appropriate description and then import the file into WDS.



Appendix B – upKeeper white label

You are now able to change images and other elements in upKeeper for branding your product. By adding images with file names ending with -custom, these files will be chosen instead of the default images. This is to make sure they are not overwritten when there is an update. Default images may be overwritten. Only the files that are specified in this document support the -custom suffix.

ADMIN WEB:

Admin web images are added to ../images/

The logo shown at login:

Default: ../images/logowhite.png

Custom: ../images/logowhite-custom.png

Recommended size: 169 x 26 px

The background shown at login:

Default: ../images/bg.jpg

Custom: ../images/bg-custom.jpg

Recommended size: 1920 x 1280 px

The logo shown in the top-bar when logged in:

Default: ../images/logo.png

Custom: ../images/logo-custom.png

Recommended size: 169 x 27 px

The favourite icon shown in the browser field:

Default: ../images/favicon.ico

Custom: ../images/favicon-custom.ico

Recommended size: 16 x 16 px, 24 x 24 px or 32 x 32 px.

Make sure to clear your browser cache after switching this logo, or the old one might be shown instead.

MY UPKEEPER:

You can change the My upkeeper text that is shown as the title of the client window, etc.

This can be done by running a registry script when installing the client. To change the title, add the key BrandingTitle to HKEY_LOCAL_MACHINE/SOFTWARE/upKeeper/Settings and change its value.

DSOS

You can change the Dynamic Service Operation System (DSOS) to display your logo and colors of your choice.

Replacing the built in Resources\logo.png allows you to edit the visible logo inside the app.

In addition, this block of settings can be added to the appsettings.json file for additional color and font customization. upKeeper defaults are used when no value is provided.

```
"Theme": { "GradientTop": "#FFCB0084", "GradientBottom": "#FFD54328", "Primary": "#FF4F0077", "Secondary": "#FF3C2A5D", "PrimaryBackground": "#FFFFFF", "SecondaryBackground": "#FFFFFF", "Text": "#FFFFFF", "TextBackground": "#FF000000", "TextBackgroundSemiTransparent": "#21000000", "FontFamily": "Georgia" }
```

To summarize the approach, we've used. A Primary Color should reflect the main color of your organization. When this color is used, it is behind the PrimaryBackground property. The

methodology is the same for all color properties.

The `TextBackgroundSemiTransparent` property refers to the banner. Only 8 character long HEX (RGBA) values are supported, if you wish for the color to be completely opaque, prefix the typical RGB HEX color with `#FF`. Values between `00` and `FF` refers to different degrees of transparency.

The `FontFamily` property is using system fonts. If you wish to use fonts that are not installed by default on windows machines, you have to install them beforehand.

Sos:

You can change the background image in the Sos:

Default: `../Sos/Upkeeper.Sos/Upkeeper.Sos.Manager/upkeeper.png`

Custom: `../Sos/Upkeeper.Sos/Upkeeper.Sos.Manager/upkeeper-custom.png`

Recommended size: 1500 x 911 px.

Sos (Debug):

Default: `../Sos/Upkeeper.Sos/Upkeeper.Sos.Manager/bin/Debug/upkeeper.png`

Custom: `../Sos/Upkeeper.Sos/Upkeeper.Sos.Manager/bin/Debug/upkeeper-custom.png`

Recommended size: 1500 x 911 px.

Appendix C – Set up Singel Sign-On

upKeeper Manager version 5.2.1 and later supports Single Sign-On (SSO). This document provides instructions on configuring and using SSO within upKeeper Manager. Configuration of external systems required for SSO is outside the scope of this document; refer to each system's documentation for specific setup instructions.

Overview

Once SSO is configured, new users can log in to upKeeper Manager by entering a *Provider Name* as specified in upKeeper Manager. This name indicates which Identity Provider (IdP) should handle the login request. The IdP will authenticate the user and, if access is granted, redirect them back to upKeeper Manager. upKeeper Manager will then allow access to the specified organization with the role of *SSOUser*.

Setup

Setting up SSO requires configuration in both upKeeper Manager and external IdP.

External IdP

Begin by configuring your IdP with a SAML endpoint and granting selected users access to it. Refer to your IdP's documentation for detailed SSO configuration. Make sure to include a callback URL pointing to your upKeeper Manager environment.

upKeeper Manager

Login into upKeeper Manager as a System Administrator (highest level). Select **upKeeper Administration** and then **Authentication Providers**. Click **Create** to add a new provider.

The screenshot shows the 'Authentication providers / MyProvider123' configuration page in the upKeeper Manager interface. The page is titled 'SAML' and contains the following fields:

- Provider:** MyProvider123
- Issuer URL:** https://{UPKEEPER_URL}/SAML2
- Endpoint:** https://{EXTERNAL IdP}/saml/authenticate/idp
- Certificate:** MIIDITCCAnECFAHOEzszTkvRaM+K91J01syjwYQMA0GCSqGSIb3DQEBCwUAMIGAMQswCQYDVQQGEwJTRTEZMBCGGA1UEC
- Client Id (ngAuthApp):** ngAuthApp
- Organization:** Default Organization

At the bottom of the form, there are 'Save' and 'Delete' buttons, and a 'Hjälp' (Help) button in the bottom right corner.

- **Provider Name** – This is the name users will use to log in via SSO to your IdP. Choose a name that is memorable yet secure.
- **Issuer** – The IdP callback URL. Replace [UPKEEPER_URL] with the URL of your upKeeper Manager web instance.
- **Endpoint** – The IdP's URL. Replace [EXTERNAL IdP] with your IdP's address. The full URL may need to be customized.
- **Certificate** – The certificate for secure communication with the IdP. Paste the certificate string provided by the IdP when configuring SAML SSO.
- **Client ID** – Specify the client ID for login. Recommended client ID is *ngAuthApp*, which is standard client id for web login.

- **Organization** – Specify the organization that users logging in with this SSO configuration will have access to.

SSO user management

After setting up SSO, you can manage users who log in to upKeeper Manager via SSO. upKeeper Manager will automatically create an account for SSO users the first time they log in. These users will be listed under **Users** in the **upKeeper Administration** section. Accessible organizations and roles can be modified per user to suit your needs.

Appendix D - Custom script during Windows Setup

SetupComplete.cmd and ErrorHandler.cmd are custom scripts that run during or after the Windows Setup process. They can be used to install applications or run other tasks. upKeeper Manager have built in functionality to support this feature.

Function

If SetupComplete feature is triggered upKeeper Manager will search for SetupComplete folder, first in the booted SOS folder structure and then on connected SMB share if available. If folder SetupComplete is found, content of the folder is copied to “%WINDIR%\Setup\Scripts\”. File SetupComplete.cmd is updated and following texts are replaced.

[ORGANISATIONID] – replaced with id of current organization.

[ENDPOINT] – replaced with the first endpoint address in appSettings.json.

[ENDPOINT1] – replaced with the first endpoint address in appSettings.json.

[ENDPOINT2] – replaced with the second endpoint address in appSettings.json.

Trigger

SetupComplete feature is triggered by adding JSON configuration in settings for a computer device. JSON settings can be added on a single computer device or inherited from group or platform. Final setting can be viewed by clicking settings button on a single computer in upKeeper Manager.

A minimum of two curly brackets ({}) is recommended if no other JSON configuration is needed. JSON configuration can be used for Microsoft Intune configuration.