LEAPWORK

Technical description

The LEAPWORK Automation Platform

LEAPWORK

Technical description: The LEAPWORK Automation Platform

Contents

1. Document purpose

This document contains a description of the LEAPWORK Automation Platform from a logical, technical, and security perspective. It describes the software components and network infrastructure that make up the technical infrastructure and describes how the security model in LEAPWORK works. It also includes a best practice guide for installation of the LEAPWORK product in an enterprise. Finally, it includes a description of the service and support model that ship with LEAPWORK.

The document can be used to gain an overview of actions and tasks involved with implementing LEAPWORK in an organization. The document is not a walkthrough of the individual installation steps but will include links where relevant.

2. Executive summary

The LEAPWORK Automation Platform lets companies create automation cases for web, desktop, virtual desktop applications, and more.

The LEAPWORK Automation Platform is designed and developed for *automation of UI based processes* and comes with the following capabilities:

- Cost-efficient and codeless creation of automation cases
- Management of large suites of automation cases
- Scheduled and manual execution of automation cases on local and remote computers
- Comprehensive reporting and visual representation of results for execution of automated cases

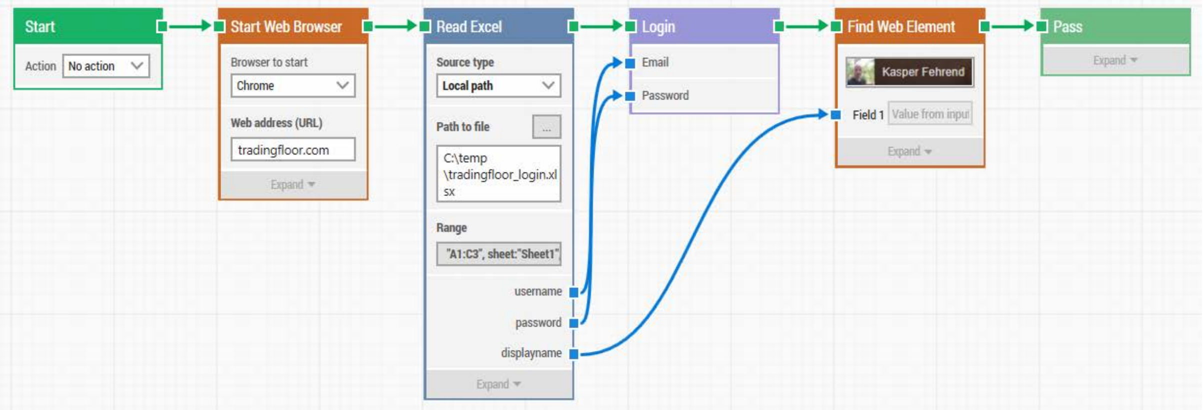
This document describes the technical aspects of introducing LEAPWORK into an organization. It covers the logical and technical infrastructure of the LEAPWORK Automation Platform and how to deal with security.

3. Introduction

The LEAPWORK Automation Platform is an intuitive platform for automation of processes in software.

The automation cases in LEAPWORK Automation Platform are designed as visual flowcharts. The flowcharts are made by connecting various building blocks. The building blocks each perform one or more operations, like “Click element”, “Start application”, “Find Web Element” etc. Using the building blocks, it is easy to transfer the domain knowledge about the system under automation into an automation case.

The image below shows an example of how an automation case in LEAPWORK could look like.



Action: No action

Browser to start: Chrome

Web address (URL): tradingfloor.com

Source type: Local path

Path to file: C:\temp\tradingfloor_login.xlsx

Range: "A1:C3", sheet:"Sheet1"

username

password

displayname

Email

Password

Field 1: Value from input

LEAPWORK supports easy reuse of sub-flows and data-driven patterns using databases, spreadsheets, web services etc. as part of the automation case design.

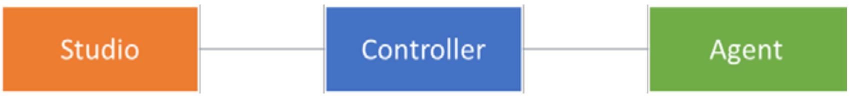
The automation cases can cover web and desktop applications (e.g Office, SAP, and Java), as well as virtual desktops (e.g. Citrix applications, remote desktops, mainframe terminals, etc.).

Remote execution of automation cases, scheduling of automation, advanced reporting and visual dashboards are all an integrated part of the LEAPWORK Automation Platform, which makes it capable of handling all aspects of functional UI automation.

4. Architecture

4.1 Elements of the LEAPWORK Automation Platform

The LEAPWORK Automation Platform consists of three major components: Studio, Controller, and Agent.



Studio is a Windows desktop application. It is a powerful, visual designer that you can use to easily create automation cases, set up run schedules, and analyze results with dashboards and reports.

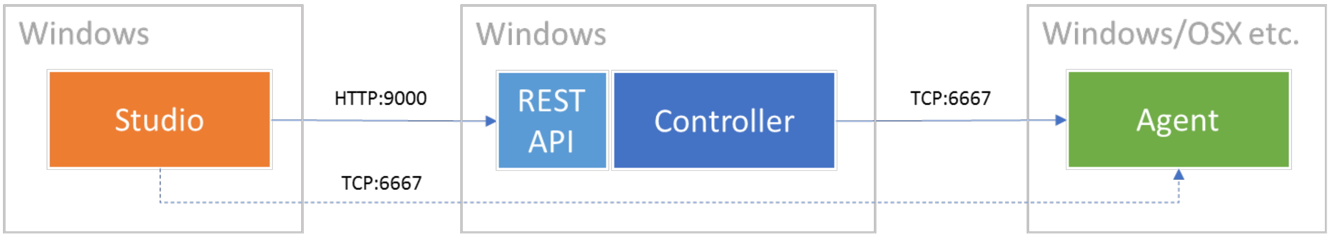
The **Controller** is the server of the LEAPWORK Automation Platform. It is a Windows service that runs in the background. The Controller hosts all automation assets, such as automation case definitions, images, scheduling configurations, reports, and more. It also controls how automation cases are run and where the cases are executed. No elements are saved locally with Studio; all Studio assets and configurations are saved in the Controller.

The last component is the **Agent**, which is a Windows service that will run in the background on any operating system or device, where the automation cases should be executed. We call these locations ***Environments***. The Agent takes care of executing the automation cases on a defined environment and will continuously update the Controller on the status of the automation cases.

The number of Agents, Studios and Controllers you have available depends on the purchased license.

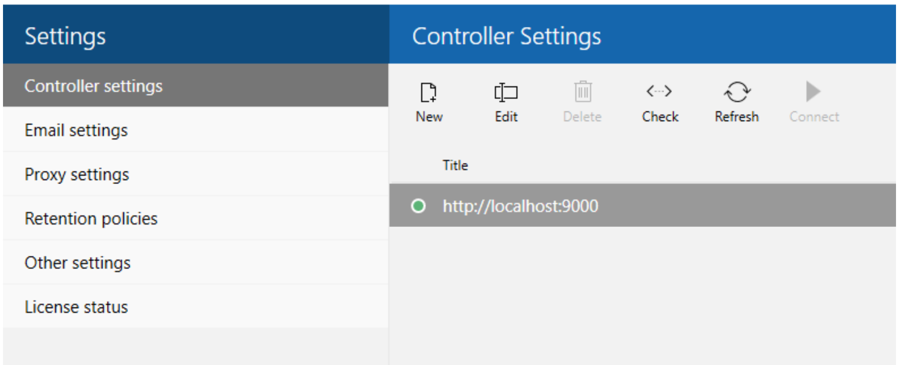
The Controller and Agents can be installed in closed networks without internet access, such as firewalled test networks. Studio, on the other hand, does require an internet connection for license activation and error reporting.

4.2 Technical overview



Studio connects to the **Controller** using HTTP on port 9000 (configurable). Studio will always initiate the connection between Studio and the Controller, so the HTTP protocol and port should be allowed from Studio to the Controller.

In Studio, it is possible to configure which Controller to connect to, by specifying the URI for the REST API and the selected port, e.g. *http://localcontroller.dom:9000*

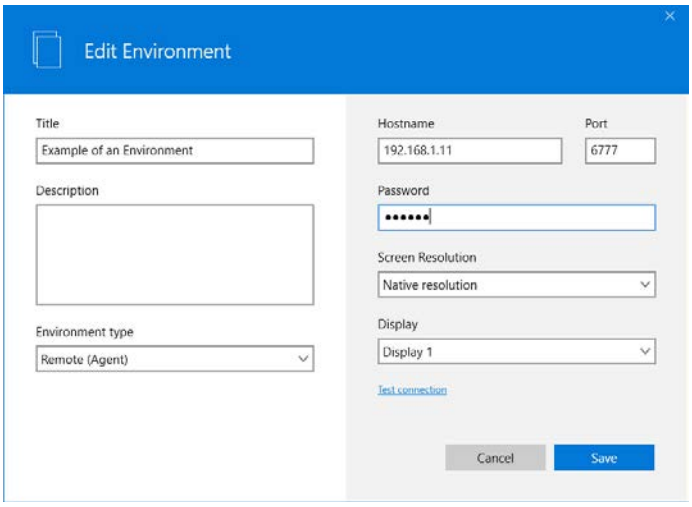


○ http://localhost:9000

As part of designing the automation cases it is possible to do a *preview run* of the cases in Studio. A preview run is like running the automation case via an Agent, but an Agent is not actually used, even if the local machine has one installed. This means the automation cases can be designed without installing an agent on the same machine as Studio.

All resources, automation case configurations, images etc. are stored as assets on the **Controller**. All assets are stored as JSON files on the file system in the Asset folder. The location of the Asset folder is configurable in the controller.config file found in the LEAPWORK installation directory (Typically, C:\Program Files\LEAPTEST).

On top of hosting all resources, the Controller is also handling the execution of automation cases on the **Environments**, or more specifically, the machines where the Agents are installed. The environments are defined in LEAPWORK and contain information about how the Controller connects to Agent service; usually the name/IP address and a port number.



Title: Example of an Environment

Description:

Environment type: Remote (Agent)

Hostname: 192.168.1.11

Port: 6777

Password: ••••••

Screen Resolution: Native resolution

Display: Display 1

Find more information about how to set up environments here:
www.leapwork.com/services/learning-center/schedules-and-reports

The Controller connects to an Agent using the information in the environment, using a closed remote-control protocol on port 6777 (configurable). It will always be the Controller that initiates the connection to the Agent, and the connection can be secured by a password. The password is specified when the Agent is installed and needs to match the password in the environment (see more in section 4.3 Security)

When an automation case is executed on an Agent, the Controller sends all necessary information to the Agent, which then executes the automation case. The Agent will continuously send screenshots and results back to the Controller using the already open connection.

It is an option – not mandatory – to allow Studio to access an Agent while designing the automation case. This can be very helpful especially for virtual desktop automation that is based on image and text recognition. To support this, Studio needs to be able to access the Agent the same way the Controller does.

4.3 Security

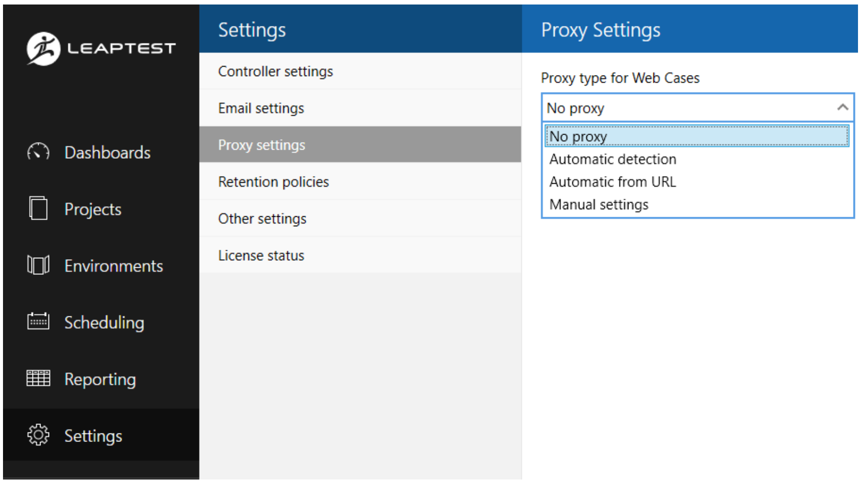
4.3.1 Studio

Part of designing automation cases in Studio is to run the cases in preview mode. As previously mentioned, it means running the automation case locally without using an Agent. In preview mode, the automation cases run with the privileges of the account that opened

LEAPWORK. This means that accessing external data sources (databases, network drives, etc.), logging in to proxies, accessing local folders, etc. are executed as if the user was doing all these actions.

The only exception to the above is when a browser is opened as part of a web automation case. LEAPWORK uses Selenium as the engine for web automation, and the browser initialized is not opened with the user's profile. Instead, it is running as 'Local System', which has no browser configuration attached. This includes lack of proxy settings, which means that there might be situations where a user can open a normal browser and navigate to a website, but when the browser is opened from LEAPWORK, the website is not available.

To handle this, it is possible to specify the proxy settings to use for the web automation cases. This is done under 'Settings' – 'Proxy settings'. These settings work both for running the cases locally in Preview mode and for cases executed using the Controller (e.g. scheduled cases).

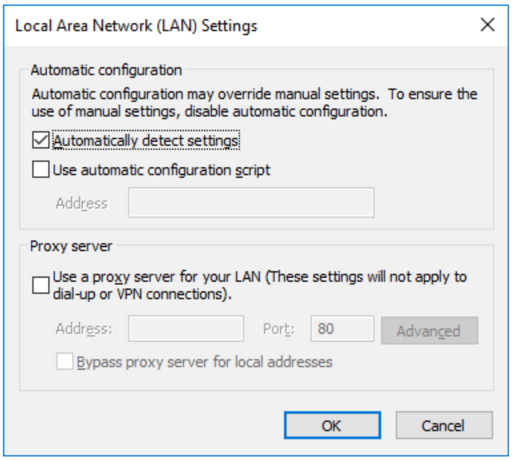


Settings

Proxy Settings

Proxy type for Web Cases: No proxy

The settings should match the settings in the local web browser.



Local Area Network (LAN) Settings

Automatic configuration

Automatic configuration may override manual settings. To ensure the use of manual settings, disable automatic configuration.

☑ Automatically detect settings

□ Use automatic configuration script

Address

Proxy server

□ Use a proxy server for your LAN (These settings will not apply to dial-up or VPN connections).

Address: Port: 80

□ Bypass proxy server for local addresses

To load plugins, bookmarks, extensions and more, it is possible to use a 'User Profile' file in Chrome and Firefox. Find more information here: https://support.leapwork.com/hc/en-us/articles/360000894951

4.3.2 Controller

The Controller is a Windows service and runs with the privileges of the account running the service.

All images, configurations, logs, etc. are stored via the Controller in the 'Assets' folder. The 'Assets' folder is located on the file system and can be configured in the controller.config file.

As part of the setup of the Controller, it is important to consider the access rights for this folder. It could be that the configurations contain data that should not be accessible outside the team working with LEAPWORK. In this case, the repository can be closed so only the service account running the Controller can get access.

On the other hand, all videos recorded during the execution of automation cases are stored in the repository, so if non-LEAPWORK users need access to the videos, then access needs to be granted.

4.3.2.1 Web automation on remote environments

When the Controller is sending an automation case to an Agent, in most cases, the automation case is executed on the Agent machine. This means that access to external resources, mapped drives, etc. are handled on the Agent machine. There are a few types of setups where the access to the external resources is handled on the Controller:

For web automation cases it is possible to create Environments that point to a Selenium Grid or to cloud based devices or instances hosted by Sauce Labs and BrowserStack. Common for all these Environments is that the automation cases are executed on machines, devices, or instances where the LEAPWORK Agent most likely is not installed. This means the "Agent" in this case is the underlying Selenium engine which does not offer access to external data sources, local command prompts, etc.

In this case, the Controller will execute the part of the flow that needs external access (reading a spreadsheet, executing a query against a database, etc.) and then send the data to the "Agent". For this, the Controller needs access to files, folders, and databases (ODBC connections) to retrieve the requested data or execute a PowerShell script.

4.3.2.2 Public REST API

The Controller ships with a public REST API. The primary use of the API is to serve as the interface between Studio and the Controller, but the API is accessible for other uses as well.

The control of who and what can access the public REST API needs to be handled on the infrastructure level. This means limiting which accounts and service accounts that can access the machine hosting the Controller/REST API.

The REST API runs with HTTP only. It is not possible to install a SSL certificate and change the protocol to HTTPS. This is because the Controller is only binding methods to the exposed end point and is not running a real web server to handle the SSL handshake. If this is needed, it is possible to install the Controller behind a load balancer or similar that can handle the SSL handshake.

4.3.3 Agent

An automation case can be run in preview mode, which is used primarily when designing the automation case. In all other instances, the automation case is executed on a machine where an Agent is installed. There are a few exceptions for web automation. See section 4.3.2.1 Web automation on remote environments.

The Agent service is running in 'Session 0' and should always be configured to run as 'Local System'. 'Session 0' does not contain any UI elements, so when a case runs, a new User session/desktop is opened, which allows for UI actions (move mouse, enter text etc.). The case will run as the account that is logged into the user session, so sometimes it is necessary to include the Windows login as part of the actual case.

4.3.3.1 Admin Approval screens

In the rare cases in which an automation case triggers an action that requires higher privileges than the user account has, a new secure desktop is opened. This secure desktop typically contains a popup requesting the user to either login as administrator or just approve the actions to be executed. The secure desktop runs as 'Local System', which has the highest privileges available, so these popups/screens can also be included as part of the flow.

4.3.3.2 Communication with Controller

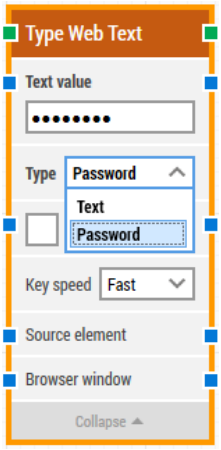
The communication between the Controller and an Agent uses a remote-control protocol on a configurable port (default: TCP 6777). The connection is not encrypted, but it is best practice in a production setup to protect the Agent service with a password. This will prevent misuse of the Agent service.

The password can be specified as part of the installation of the Agent or be configured directly in the *agent.config* on the machine hosting the Agent service. The file can be found in the *LEAPWORK\Agent* folder under the installation path – typically *c:\program files\LEAPWORK\agent\agent.config*. The same password should be specified in the environment defining the Agent.

4.3.3.3 Password handling

Automation cases often start by opening applications, websites, etc. where a login is required. This raises the issue of password handling and security.

LEAPWORK ships with a simple mechanism for securing passwords for e.g. login forms. In the building blocks that type text ("Type Text", "Type Web Text", and "Set UI Element Value") it is possible to set the Type property to "Password". This means that the text specified in the Text value field – the text that will be inserted – is shown as dots only. This way, the password cannot be picked up by viewing the automation case or the log from the automation case.



Text value: ••••••••

Type: Password

Key speed: Fast

Note that the password is stored in text form in the automation case configuration which resides in the Assets repository on the Controller, so it is very important to control the access to this repository. (See section 4.3.2 Controller.)

LEAPWORK does not contain an encrypted wallet or secure-key feature where passwords can be stored. This is a design decision based on experience and feedback from clients and partners.

If a high level of security is needed, most companies with a strict password policy have already implemented a secret server that secures all the crucial passwords used in the company. These servers usually have an API (database, web service, or script-based) that can be accessed from LEAPWORK using the standard data-driven building blocks available.

5. System management

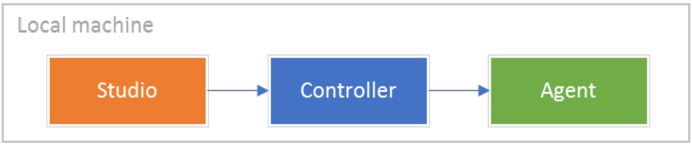
This section describes how to manage the LEAPWORK Automation Platform from an operational point of view. This includes initial installation, different proposals of installation plans, upgrade scenarios etc.

5.1 Installation

5.1.1 Possible setups

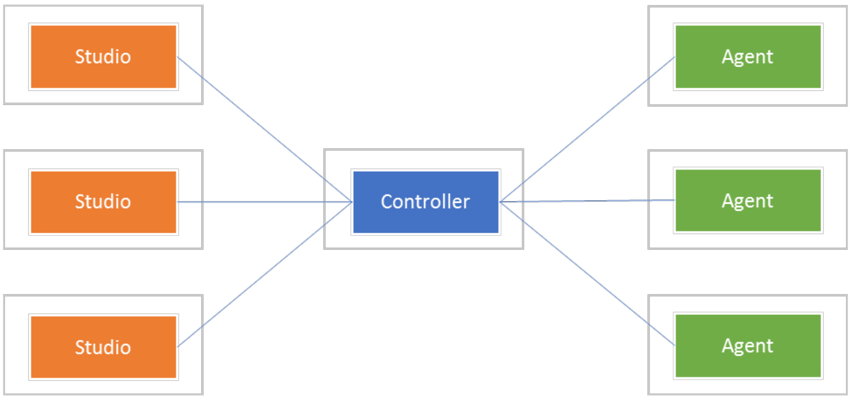
5.1.1.1 Local setup

A local installation of LEAPWORK is typically done as part of the evaluation of LEAPWORK. In a local installation all LEAPWORK elements (Studio, Controller, Agent) are installed on the same machine, and the automation cases are executed locally.



5.1.1.2 Distributed setup

Installing LEAPWORK in a distributed setup supports easy collaboration and remote execution of automation cases. The image below shows a typical setup of a distributed LEAPWORK system.



All entities can be installed on-premises or in the cloud and can be hosted on both physical and virtual machines. Please see section 4.2 Technical overview for infrastructure requirements.

5.1.2 Download of software

All the software needed to run the LEAPWORK Automation Platform can be downloaded from https://account.leapwork.com/downloads. No login is required.

5.1.2.1 Windows

LEAPWORK for Windows can be downloaded for 32- and 64-bit versions in either a .msi package or as a .zip file (containing the .msi package).

The .msi package contains all elements of LEAPWORK: Studio, Controller, and Agent, so only one download is needed for a full setup.

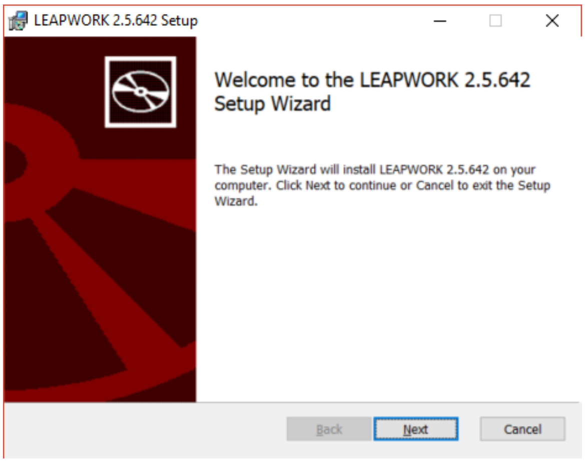
5.1.2.2 MacOS

For MacOS (Agent only) a .pkg download is available from the download page.

5.1.3 Installation of components

After downloading the latest appropriate version (32-bit or 64-bit), you install LEAPWORK Studio by double-clicking the downloaded .msi file.

1. The first screen will display the version number and ensure that the downloaded version fits your computer. Click "Next" to begin the installation.

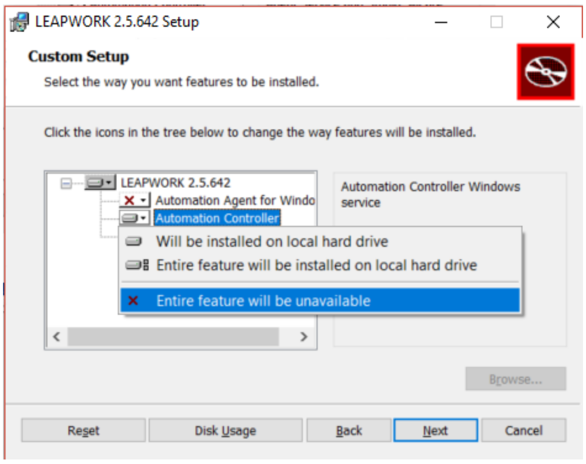


LEAPWORK 2.5.642 Setup

Welcome to the LEAPWORK 2.5.642 Setup Wizard

The Setup Wizard will install LEAPWORK 2.5.642 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

2. Choose which of the three components to install on the computer. Do this by clicking the drop arrows and choosing the appropriate option for each component. In the example below, only Studio will be installed on the local computer.



LEAPWORK 2.5.642 Setup

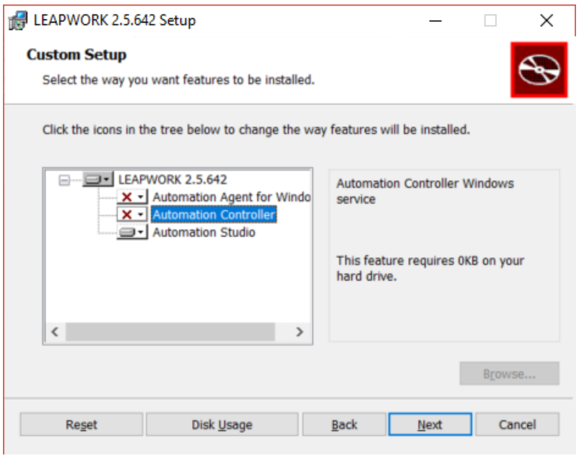
Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

LEAPWORK 2.5.642
Automation Agent for Windo
Automation Controller
Will be installed on local hard drive
Entire feature will be installed on local hard drive
Entire feature will be unavailable
Browse...
Reset
Disk Usage
Back
Next
Cancel

Automation Controller Windows service



LEAPWORK 2.5.642 Setup

Custom Setup

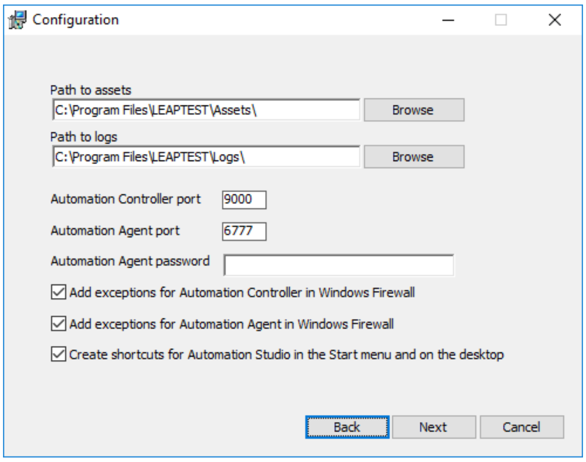
Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

Automation Controller Windows service

This feature requires 0KB on your hard drive.

3. The Configuration screen contains properties used for one or more of the components. All the settings can be changed afterwards in the config files for the individual components: ‘studio\studio.config’, ‘controller\controller.config’ and ‘agent\agent.config’. The files are in the installation directory – typically ‘c:\program files\LEAPTEST\’.



Path to assets
C:\Program Files\LEAPTEST\Assets\

Path to logs
C:\Program Files\LEAPTEST\Logs\

Automation Controller port 9000

Automation Agent port 6777

Automation Agent password

☑ Add exceptions for Automation Controller in Windows Firewall

☑ Add exceptions for Automation Agent in Windows Firewall

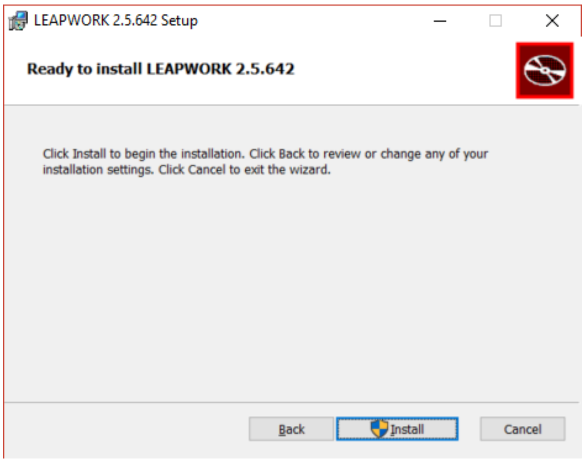
☑ Create shortcuts for Automation Studio in the Start menu and on the desktop

a. **Path to assets.** This is the path to the file system folder that will store all LEAPWORK assets, including automation cases, schedules, and run results.
b. **Path to logs**. This is the path to the file system folder, which will contain all the diagnostics logs for each of the components. Though it may never be necessary for you to examine these logs directly, LEAPWORK Support may request these logs for analysis.
c. **Controller port**. The Automation Studio and the Controller always communicate using REST API calls, and this requires the use of a TCP port. The default is port 9000, which you can change to any other port number that is not in use by other software on your computer. NOTE: Unless it is necessary for any particular reason, we do not recommend changing either the Controller port or the Agent port numbers.
d. **Agent port**. Automation cases run in the Agent, which communicates with Studio and the Controller through a closed remote-control protocol on a specific TCP port. The default port is 6777, which you can change to any other port number not in use by other software on the computer. NOTE: Unless it is necessary for any particular reason, we do not recommend changing either the Controller port or the Agent port numbers.
e. **Agent password**. Because the Agent uses a remote-control protocol, we advise you to set a password to prevent unauthorized parties from connecting to it.
f. **Add exceptions for the Controller in Windows Firewall.** A check in this box will direct the installation process to add a rule to the Windows Firewall to permit all incoming connections to the Automation Controller. NOTE: Because the LEAPWORK Automation Platform is also often used in corporate and already secure test networks, the password is optional.

g. **Add exceptions for the Agent in Windows Firewall.** A check in this box will direct the installation process to add a rule to the Windows Firewall to permit all incoming connections to the Agent.

h. **Create Studio shortcut in Start Menu**. A check in this box will direct the installation process to add a LEAPWORK Studio shortcut to the Windows Start Menu. This shortcut makes it easy to find and start the application at any time.

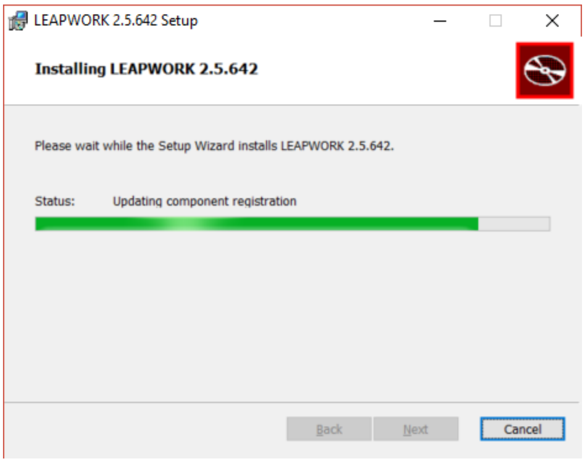
4. Click "Next" on the configuration screen.
5. Click "Install"



Ready to install LEAPWORK 2.5.642

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

6. The selected components will then be installed.

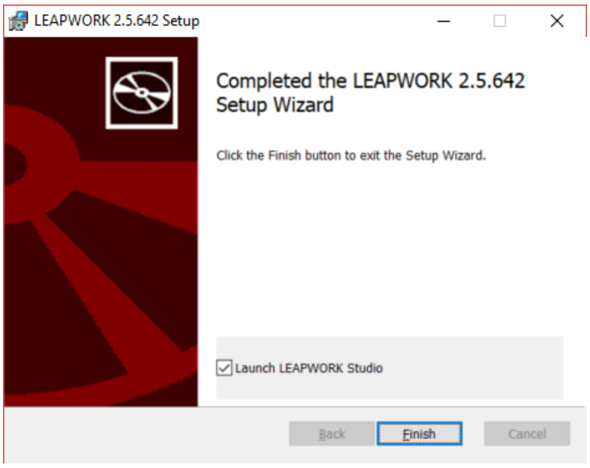


Installing LEAPWORK 2.5.642

Please wait while the Setup Wizard installs LEAPWORK 2.5.642.

Status: Updating component registration

7. After the installation is complete, click “Finish” to exit the installer.



LEAPWORK 2.5.642 Setup

Completed the LEAPWORK 2.5.642 Setup Wizard

Click the Finish button to exit the Setup Wizard.

☑ Launch LEAPWORK Studio

5.1.3.1 Installation of Agent on MacOS

It is possible to install the LEAPWORK Agent on OS X / MacOS. Please refer to the following link for installation instructions: https://support.leapwork.com/hc/en-us/articles/115001200692-How-to-install-the-Agent-on-Mac-

5.1.4 System requirements

All three components of the LEAPWORK Automation Platform can be installed on the following versions of Windows:

- Window 7 (update 1), Windows 8.0 + 8.1, Windows 10
- Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016

The OSX version of the LEAPWORK Agent can be installed on OSX 10.8 or newer.

Typically, Studio and the Agents are installed on the desktop versions, and the Controller is installed on a Windows server. (See below.)

Studio should be installed on the user’s work station. Studio is comparable to a normal Office installation, so in most cases no special configuration or attention is needed.

Updates of LEAPWORK are available from the download page: https://account.leapwork.com/downloads. To install a newer version, download the .msi file and follow the instructions in section 5.1.3 Installation of components.

Be aware that all installed components of LEAPWORK need to be of the same version to be able to connect. This means that the upgrade should include all installed instances of Studio, Controller, and Agents.

Part of the installation of a newer version is an automatic upgrade of the existing assets. The assets are all stored in the Asset folder on the Controller machine.

The **Controller** is the central hub in the LEAPWORK system and needs to be configured to accommodate the workload. The workload depends on various factors which can all influence how the Controller machine should be configured. These include:

- Number of automation cases
- Complexity and duration of automation cases
- Frequency of execution of automation cases

The recommendation is to start out with a standard server configuration, monitor the Controller machine on memory usage, I/O activity, CPU usage etc., and then down- or upgrade the machine. A standard server configuration could be:

- 16 GB RAM
- 4-6 cores
- Flexible and fast disc system (SSD, SAN, or similar)

The Controller can run on both physical and virtual servers.

The **Agent** is used for executing automation cases which can make the configuration of the machine move in two directions:

- Simulating applications for an end-user:
    - In this case, the machine should be as similar as possible to the machines the end-users will be using. This will increase the reliability of the automation cases.

- Running automation cases in parallel:
    - In web automation, it is possible to run browsers in parallel (using a Selenium Grid) on the same machine. The more powerful the machine, the more browsers can run in parallel.

5.2 Upgrades

Updates of LEAPWORK are available from the download page https://account.leapwork.com/downloads. To install a newer version, download the .msi file and follow the instructions in section 5.1.3 Installation of components.

Be aware that all installed components of LEAPWORK need to be of the same version to be able to connect. This means that the upgrade should include all installed instances of Studio, Controller, and Agents.

Part of the installation of a newer version is an automatic upgrade of the existing assets. The assets are all stored in the Asset folder on the Controller machine. (See section 4.3.2 Controller.)

5.3 Logging

All three components in LEAPWORK write to individual log files that can be found in the "Logs" folder in the installation directory (typically "c:\program files\LEAPTEST\Logs"). All the log files are automatically archived each day, so there will be one file for each component per day.