MID

the modeling company

Innovator 15.1

Installation Manual

innovator

Installation Manual – Innovator 15.1 – Edition November 2021

Contents

Brief Description

Who Should Read This Information?

Everyone who wishes to install Innovator version 15.1, particularly teams in networks.

Purpose of Information

The information describes the worksteps necessary for installing Innovator 15.1.

This information will help you when installing Innovator, particularly when installing in heterogeneous networks; it is also designed to help you avoid technical issues when working with Innovator 15.1.

Initial Installation, Upgrade, Update and Component Exchange

You must take various aspects into consideration when installing Innovator.

- Initial Installation

  First time installation requires a precise analysis of the requirements you expect when using Innovator so that decisions about architecture can be made. Architectural modification may require a lot more time.

  MID GmbH's order processing department needs to release the license repository.

- Upgrade

  An upgrade refers to changing from one version to the next (e.g. 15.0 To 15.1). Installation is carried out in parallel to any previous installations; it is also possible to simply make architectural changes.

  The license repository of the new version needs to be released by the order processing department.

- Update

  An update refers to a software update where the major and minor versions are still the same (e.g. 15.1.1 to 15.1.2) The previous architecture will not be modified.

  The license repository is adopted and no extra license release is required to continue working with Innovator.

- Hotfix

  If the build number has been changed in a release, this shows the presence of a hotfix. Hotfixes update the existing installation and are compatible with the interface.

  Server hotfixes update the server to fix problems which are usually customer-specific. Previous client applications are compatible with these hotfixes.

  The license repository is always maintained and operation can continue.

- Component Exchange for Server

  It is possible to switch out single files causing issues (e.g. DLLs) with corrected versions to solve problems which are normally customer-specific. Extreme caution must be exercised when carrying out such a switch.

  Previous client applications and other server components are compatible with these files.

  The license repository is always maintained and operation can continue.

Hardware and Software Basis

Innovator clients can be run on a PC with 2 GHz or more with Microsoft Windows 8.1 or higher and the respective current service pack.

Windows Installer 5.0 is required for Innovator installation.

Supported Platforms

Innovator runs with the following hardware configurations:

- Server (64 bit)
    - PC with 3 GHz or more with Microsoft Windows 8.1 or from Windows server 2012 R2 with the respective current service pack
    - PC min. 3 GHz with openSuSE 12.3 or higher
    - PC min. 3 GHz with Red Hat Enterprise Linux 7.2 or higher
- Client (64 bit)

    PC with 2 GHz or more with Microsoft Windows 8.1 or higher with the respective current service pack

The installation of the following is also a system prerequisite:

- Visual C++ Redistributable Packages for Visual Studio 2015-2019 (64 bit)

The required runtime environments for Java 11 and .NET Core 3.1 are included within the scope of delivery and installed as an integrated part of Innovator.

For custom extensions (engineering actions, plug-ins), both a Java API and a .NET API (e.g. C#) are an integral part of the Innovator installation; no separate download is required.

Hard Disk Space Requirements

Innovator requires a hard disk space of 3 GB.

Memory Requirements

- Innovator requires at least 2 GB of available memory on a single station, but 4 GB is recommended
- Small teams require at least the following: Client 2 GB; server 2 GB, recommended 4 GB
- Enterprise teams require at least the following: Client 2 GB; server 4 GB (approx. 1 GB per model)

Graphical User Interfaces

A screen resolution of at least 1600 x 900 pixels and a graphics memory or 1 GB is recommended for ease of use.

Getting Started and Help

Getting Started and Help (offline and online) require a browser with support for HTML5.

Innovator Architecture

Innovator uses a client/server architecture.

Innovator is designed as a client/server system. All data is collected in data pools (repositories); clients do not have direct access to these. Administration of a repository falls to a single server program respectively. This model server ensures the following:

- Consistency of managed data within models
- Mutual exclusion of users for exclusive editing of elements
- Central management of users and access rights

Client applications access the model server via the network. Data exchanged via telegrams is shown in the model editor and enables creation and editing of model elements and diagrams.

> **Note**
>
> Further architecture elements are required for communication in the network. A bus (inobus) is responsible for the communication of messages between the system components. The system environment which the model servers use for running Innovator is monitored and managed by an agent (inoagent).

Architecture Elements

Innovator architecture elements enable efficient and safe collaboration in a team.

License Server

The main license server is the server architecture for Innovator. It performs the following tasks.

- License Management
  As the name suggests, these components manage the licenses required for use of Innovator and distribute the licenses to client applications and users upon login. You can install Innovator if you do not have a valid license, but functionality will be reduced to demo mode.

  The main license server can distribute the license pool to subordinate internal or external project license servers to enable organizational separation of various areas.

- Central User Management
  When the main license server is in standard single sign-on mode, it is used for managing all users and their access rights to models, as well as administrative functions.

  You can use the Lightweight Directory Access Protocol (LDAP) to automatically update users thanks to the synchronization feature with company-wide user directory.

- Central Starting Point and Model Directory
  When the license server is started, every model server communicates certain significant regarding the repository and its models. As a result, the license server can then forward this information to the client applications. When a user selects a model, the license server coordinates the initial contact between model server and client application. The actual data traffic between the two computers is then carried out directly. Even so, contact to the license server is maintained on both sides.

> **Note**
>
> All Innovator application components need to be able to establish network connections to the respective license servers. Network port release is required,

Model Server

The model server provides model data for Innovator client applications.

It uses parameters to read the repository and uses this data to build an in-memory database. Once logging-in to the license server, the model server can be accessed for connection via the client applications. It stops multiple users accessing model data at the same time and maintains consistency.

The model server and client applications communicate used zipped telegrams which contain both commands and information.

Repositories and Models

The database that contains all model data and profiles is known as a repository.

Repositories up to version 14.1 with their own independent model server running on the main license server can be used as corporate repositories by all internal project license servers.

Innovator can be used to manage model versions. This makes it possible for the same model to be used in various states. Central deployment is not available for managed models.

Any number of users can always work on a model at the same time. The Innovator model is equipped with a locking concept to ensure that an element cannot be changed by more than one user at the same time. This allows a user to exclusively reserve elements and diagrams for modification. Other users continue to be informed of the current state of the locked object, but cannot actually make changes to it at that time.

Bus

The bus is a communication component which enables license servers, model servers, client applications and agents to quickly and asynchronously exchange status information.

A bus exists for each license server. It is also automatically started when a license server is started.

> **Restriction**
>
> The bus (inobus) also needs to be set up as a separate service when using a service to start the license server on Windows.

Hub

Each license server automatically starts a hub (inohub), also as a service. A hub provides central directory services. It is used for e.g. name resolution for alternative Innovator links to physical models.

Agent

The agent is a management component that controls and manages the system environment for the servers used for Innovator operation. Its main task is to maintain and manage all models in the version control.

Agents (inoagent) use Apache Commons Daemon as service manager. You can set them as services.

Client Application Model Editor

The model editor is the tool modelers use to access models and create and edit model content. This program is only available for Windows and an application based on Microsoft's "WPF" architecture.

It contacts the set license server upon starting the model editor and provides the models available for the user to log-in to. The functions provided by the model editor depend on which license is available for the user, which configuration settings are stored in the repository for the model and which role the user is logged-in as.

You can add many plug-ins to the model editor to add extra features. Plug-ins for a wide variety of functions can be both provided by MID and configured by the client themselves.

Further Client Applications

The configuration editor is used for editing UML profiles which form the basis of models. Profiles maintained by MID now include customization profiles. These can be used to add customer-specific extensions. Model documentation and model editor display settings are made in the configuration editor.

The main tasks of the administration program are to manage licenses, servers, models, model versions, users and login rules. You can also manage the environment and Windows services here.

The transformer is used for transforming repositories and model versions into a new Innovator version.

Network

A network is always used for component communication, unless Innovator is run solely on a stand-alone computer. Innovator communicates with its own log based on TCP/IP (Transmission Control Protocol/Internet Protocol).

Innovator currently only supports IPv4 networks. Components need to be able to build direct connections to each other; proxy servers and NAT firewalls are not supported.

Communication ports need to be open for logging communication on computers which the license server, bus, agent or model server is running on.

> **Note**
>
> As client applications require frequent and intensive communication with the model server, there are particularly high demands set on the latency of the network connection. Usability of Innovator may be greatly restricted if accessing e.g. via an Internet VPN connection.

There is no limit to either the number of clients or repositories in the network. Any number of model servers can be started on an Innovator network and these can be accessed by any number of users.

On entering the interface editing mode, the user is given the opportunity to work on any of the repositories for which an available model server has been started, and the models contained therein. Irrespective of the physical location of the data in the network, in principle, this enables every Innovator user to work with every model.

User and authorization management in Innovator makes it possible to control access to individual models and the authorizations in them.

Managing the Architecture

You can specify important Innovator architecture properties upon installation which set certain environment variables.

Architecture Specifications

The potentially complex architecture of Innovator means that a small amount of input is required to define individual structures. This input needs to be carried out upon installation. However, this can also be easily changed in the **Administration Program** at a later stage.

Different operating systems use specific techniques for storing these settings.

- The registration settings are stored in the tab under Microsoft Windows.
- Under Linux the individual settings are assigned to environment variables.

The following talks about environment variables in general.

Some settings, pertaining to the following questions, are absolutely necessary:

Which computer will the main license server run on?

This setting is of critical importance for operating Innovator. Due to its central position in the Innovator architecture, the main license server must be accessible from every computer that will run Innovator programs.

The computer that will run the main license server must be determined during installation. Although it is possible to change it at a later date, this would require all Innovator licenses to be activated by MID GmbH again.

Should the installation be carried out for the current user or for all users?

This specification determines which start menu directory the program group will be stored in and where the Innovator entries will be stored in the registration.

The Innovator server can only be set as services if the entries are made for all users on the local computer.

Where should programs, settings and repositories be located?

Specifying installation directories determines where the executable programs, common data and settings will be available. The settings contain the directories for repositories and parameter files. These directories can be modified in the **Administration Program** once the installation has taken place. This is particularly significant if working with network drives.

> **Note**
> If you are using the so-called "Services" under MS Windows, then project directories with repositories should not be located on network drives.

Server and Port Ranges

The model server contacts the license server in regular intervals. However, the license server may also call the model server to determine whether it is still running.

For communication between model servers and clients, the range of port numbers can be derived from the license server's port number specified upon installation and according to the maximum amount of concurrently running model servers.

> **Note**
>
> The license server's port number must be a value between 1024 and 64000

P is the port, which is fixed for the license server computer (e.g. 15100). The model server tries to connect to the p+1 port (e.g. 15101). If this port is already assigned, the consecutive one is checked until a free port is found. Because of this, a possibly existing firewall between license server, (for the administration program (`InnoAdministration.exe`) main and project license server, if applicable), model servers and clients has to allow all IP connections from and to the license server's port (e.g. 15100), as well as all packages in the port range above for communication between model servers and clients.

A certain port number can be set for each model server for opening the connection.

If you use Linux as your operating system and frequently start up and shut down model servers, you should use a few other ports. This should be done for security reasons, since Linux only releases ports after a time specified by the Linux administrators (up to 30 minutes).

Port Range for Main License Servers and Project License Servers

If you need to manage projects in different Innovator versions, always use one project license server for each Innovator version (##.#). One effect of this is that you do not need to change the port assignment and services based upon this for these project license servers in the case of an upgrade (version change). Furthermore, the port of the main license server remains unchanged. Only the new project license server of the new Innovator version then requires a port definition.

Communication between Server and Client

Innovator requires Message Queues for communication between the Server and the Client (e.g. Oracle and MySQL).

The default on most operating systems is too low for message queues, particularly when a lot of clients are using it. If this is the case, an error message "cannot create message queue" often appears.

We recommend increasing the value of the message queue to at least 256.

You can verify the current value with the command `ipcs -ql` when using **Linux**.

Input the number of available message queues in the `/etc/sysctl.conf` file:

`kernel.msgmni = 256`

The new value will be adopted upon reboot or directly by inputting `sysctl -p` in the console.

The amount of message queues being used in the current system should be verified after a certain amount of time using the `ipcs -q` command (in Innovator or databases). Adjust the value in such a way that it offers adequate reserves.

`ipc` does not alter the message queues of processes which have been unexpectedly ended. If necessary, these can be engaged with the `ipc` command so that the message queues can be used again.

Scenarios for Agents, License Server and Project Directory

Useful Scenarios

* **Common (Project) License Server and Common Project Directory**

  Multiple agents run on various computers with common license server and project directory.

  Model versions (model server) can be started on various computers to spread the load.

  All agents (e.g. in the physical view) show all model versions. It is clear where they are running.

* **Common (Project) License Server and Separate Project Directory**

  Multiple agents run on various computers with common license server and separate project directories.

  Repositories can physically be on various hard drives if e.g. there is not sufficient disk space.

  Agents provide different model versions. It is clear where they are running.

Problematic Scenarios

* **Agents with Different Project License Servers and Common Project Directory**

  Multiple agents are running with different (project) license servers (of the same version) but with a common project directory.

  You cannot determine which agent is responsible for the model versions.

  You cannot create model versions or these are misassigned.

* **Agents and model servers with different project license servers and common project directory**

  Multiple agents are running with different (project) license servers (of the same version) but with a common project directory.

  Agent and model server are run on different buses and do not come into contact with one another.

  Model version states cannot be identified as unique. This may lead to mistakes being made.

Architecture and Configuration Example

Example architecture to show how the components involved interact with one another in an Innovator installation.

Constellation

The architecture example illustrates various aspects of how system components and various setup options interact with one another. It is merely a suggestion of how to install Innovator. (You can find installation examples for concrete scenarios in the "Installation Examples" chapter.)

Architecture and Configuration Example Elements

* SRV01: "Server" Installation Type (for the Central Main License Server)
* SRV02: "Server" Installation Type (for the Model Server)
* SRV03: Manual Linux Installation
* Client01: "Client" Installation Type
* Client02: "Complete" Installation Type
* Network Drive with File Share
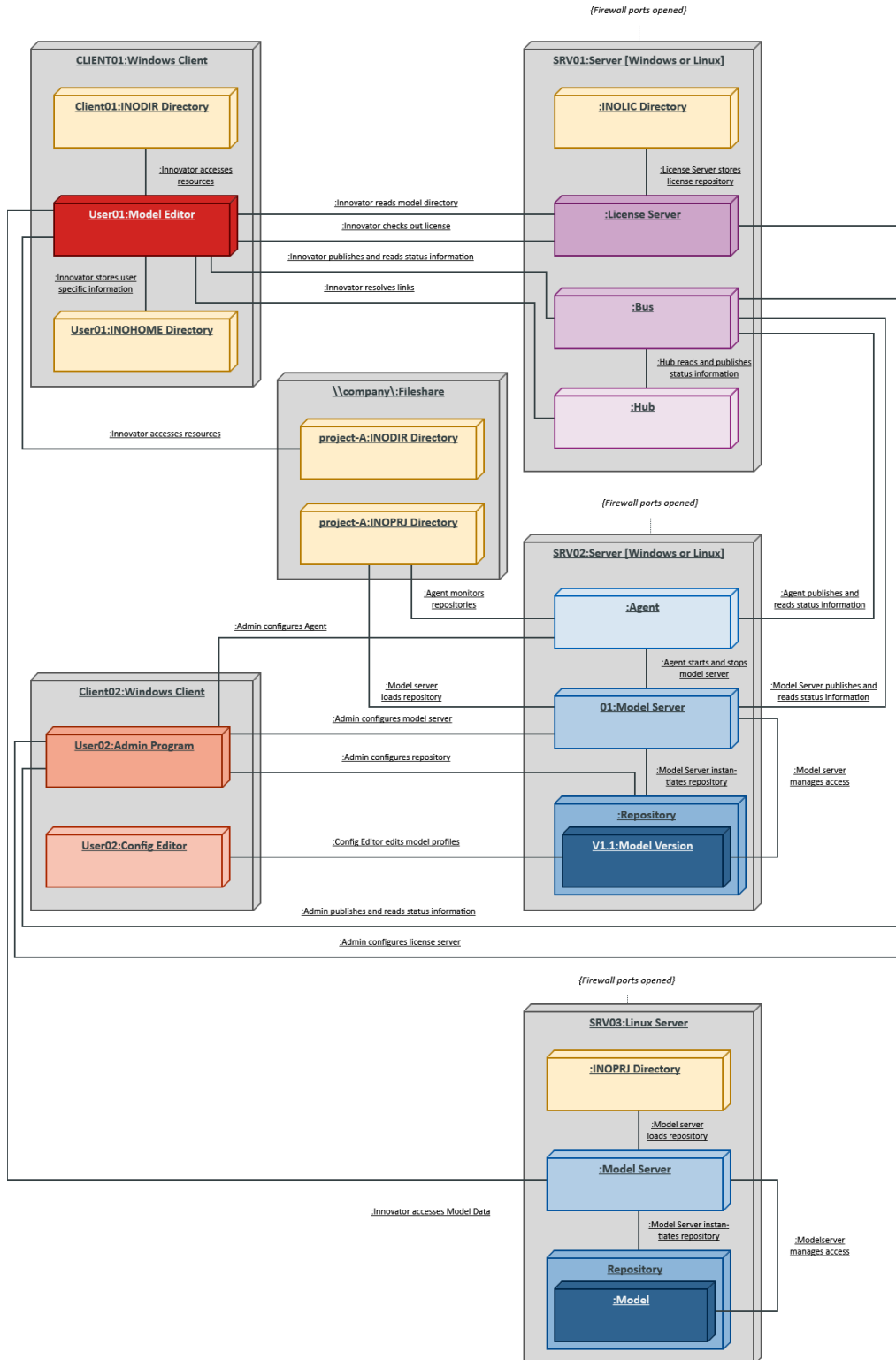
Architecture and Configuration Example

Software Installation

The Innovator installation routine needs to be executed on every client or server computer (Windows).

Predefined installation variations can be selected in the installation program. You can also individually select components you which to install.

You need to manually carry out the installation on Linux computers as per the manual. Client programs can only be run on Windows computers.

Relevant Environment Parameters

The Innovator system configuration is primarily determined by environment parameters. These point to directories or computers that are important for function as requested by the system.

(You can find an overview of these parameters in the "Table: Overview of Environment Variables" chapter.)

SRV01 Computer

This Windows computer serves as the central license server computer and plays a prominent role in the Innovator environment.

This computer should always be accessible and should be the last computer to be shut down and the first to be started upon system reboot.

Example values for environment variables:

- INOHOST: SRV01.15100
- INODIR: C:\ProgramData\Innovator\15.1 (Installation default)
- INOLIC: D:\Innovator Data\15.1\inolic
- INOHOME: not relevant
- INOPRJ: not relevant

SRV02 Computer

Model servers for managed models should be run on this Windows computer.

The models used can be found under version management and should be started by a network drive to enable central backup system.

Example values for environment variables:

- INOHOST: SRV01.15100
- INODIR: C:\ProgramData\Innovator\15.1 (Installation default)
- INOLIC: not relevant
- INOHOME: not relevant
- INOPRJ: \\company\project-a\15.1\inoprj

SRV03 Computer

Independent model servers should be run on this Linux computer.

Repository data is backed-up based on the computer's local file system.

Example values for environment variables:

- INOHOST: SRV01.15100
- INODIR: /usr/innovator/15.1/inodir (Installation default)

- INOLIC: not relevant
- INOHOME: not relevant
- INOPRJ: /data/innovator/15.1/inoprj

Client01 Computer

Innovator users model on this computer (type). Only the model editor is installed on these computers and no other administrative client programs.

You will not be able to start local model servers if server components are missing, even if you have sufficient rights to do so.

Example values for environment variables:

- INOHOST: SRV01.15100
- INODIR: C:\ProgramData\Innovator\15.1 (Installation default)
- INOLIC: not relevant
- INOHOME: C:\Users\<user>\AppData\Roaming\Innovator\15.1 (Installation default)
- INOPRJ: not relevant

The INODIR directory can also be shown on a central drive if company-specific settings should be managed centrally. In this example, that would be:

- INODIR: \\company\project-a\15.1\inodir

Client02 Computer

This computer is used for administration and configuration. Complete installation is carried out on this computer so that all relevant scenarios can be tested. All client programs are installed, but it is also possible to start a model server for test purposes.

Example values for environment variables:

- INOHOST: SRV01.15100
- INODIR: C:\ProgramData\Innovator\15.1 (Installation default)
- INOLIC: not relevant
- INOHOME: C:\Users\<user>\AppData\Roaming\Innovator\15.1 (Installation default)
- INOPRJ: not relevant

Settings

> **Note**
>
> The settings (e.g. INODIR) are identical for all operating systems and are used together.
>
> However, note that path specifications are operating-system-specific and must be set accordingly in the respective environment variables.

Installation Basics

Information for installing clients and servers in the network.

Server, Client and Repository

Innovator is available server-sided for various Windows and Linux operating systems. These Innovator versions can be integrated in a heterogeneous network and operate with identical data. In the process, the interaction and storage of the individual products can be handled in an extremely variable manner.

Client interfaces are only available for Windows versions.

Innovator keeps the data in a standard online repository, which is described with an open data model so that all of a project's data can be exchanged between all systems as desired.

Installation Files

Download

You can download the installation files for Innovator from the MID website after registering either as a potential buyer or as a customer.

Registered customers with service agreements have access to the exclusive customer area, which contains all installation files for all operating systems and versions supported. You can find information about the products and installation in a separate document (PDF) and in this Help.

Please read the Help topic "Installing an Update" to find out the differences between version migration (upgrade) and new release (update).

Installation Files for Client and Server using Windows

The installation files for direct or administrative installation of the client and server are available in German and English. Language selection determines the language for installation dialogs and the default interface language setting. You can use the corresponding environment variable to change the interface language at a later stage.

Installation Files for Servers using Linux

The installation files for the various Linux variations are language neutral. They can only be used to install the server. They use an installation script where you can set the interface language using the appropriate environment variable.

Offline Help can be Separately Installed

If you do not have access to the online help (http://help.innovator.de) then it is a good idea to install the offline help locally.

The offline help installation package is available to registered customers with a maintenance contract in the exclusive customer area.

The offline help is stored in the $INODIR\help directory as default. You need to select it as a help source in the **Options** under **Resources** to be able to access it.

Installation Planning

> You must take in all important requirements for setting-up and running Innovator when planning the installation.

Architecture Specifications

The potentially complex architecture of Innovator means that a small amount of input is required to define planned structures. This input needs to be carried out upon installation. However, this can also be easily changed in the **Administration Program** at a later stage.

Different operating systems use specific techniques for storing these settings.

- The registration settings are stored in the tab under Microsoft Windows.
- Under Linux the individual settings are assigned to environment variables.

The following talks about environment variables in general.

The following table provides an overview of the most important settings defined during the Innovator installation. The first column lists some expressions that are frequently used in the installation programs and in this introduction, the second column lists the environment variables that contain the relevant data. The third column contains further explanations.

Table: Overview of Environment Variables (Extract)

Meaning	Variable	Description
Interface language	INOLANG	Language in which to work with Innovator. The installation package language is used as default.
License Server Computer	INOHOST	Computer on which the license server is to run and details of the communication channels in the format <License server host name>.<port number>
Settings (parameter settings)	INODIR	Directory path that the parameter files are or will be stored on; icons and the Java files for engineering actions etc. can also be found here. You can use a shared workgroup directory and, in doing so, use common settings.
License directory	INOLIC	Directory path where you will find or can store license repositories as standard. Only relevant for computers with a license server.
Project directory	INOPRJ	Default directory path component which repositories (projects) are maintained and stored in. Only relevant for computers with a model server.
User-specific settings	INOHOME	The parameters in the INODIR directory are intended as central specifications. The individual user can override these specifications using their special settings. These are stored in the INOHOME directory. If this variable is not explicitly set, then a subdirectory in the %APPDATA% directory specified by the operating system is automatically used.

Further specifications are possible and/or necessary. They are described in the installation chapters for the respective operating system type.

Some of the environment variables do not have to be set on all operating systems.

Depending on the operating system and type of installation, you require administrative rights on the respective system when installing Innovator.

Server Computer

How many servers and what sizes are needed?

It can be challenging to specify a suitable dimension for the server regarding CPU and memory, particularly when installing Innovator for the first time. Utilization of resources correlates to the utilization of the model server. This means that only rough approximations can be identified at the beginning.

An important factor is how many models you want users to have online access to. e.g. You need to provide four model versions at the same time if you have a previous version, a development version and a test version for a productive model.

Memory requirements per model depend on the size of the models, i.e. the amount of model content. This may range from a very small model of 300 MB to a very large model of 2 GB.

We recommend that you calculate a logical CPU core for each active model server. Requirements for processing power for a single processor core are mainly calculated based on the number of users who work on a model at the same time.

Which computer will the main license server run on?

This setting is of critical importance for operating Innovator. Due to its central position in the Innovator architecture, the main license server must be accessible from every computer that will run Innovator programs.

The computer that will run the main license server must be determined during installation. Although it is possible to change it at a later date, this would require all Innovator licenses to be activated by MID GmbH again.

There are relatively few system requirements for operating a license server and bus and hub components so that you do not require a powerful computer for this.

> **Restriction**
>
> 🚫 You may run into issues when running the main license server on a virtual machine as computer identification may change if changes are made to the virtual machine. This, in turn, would cause issues with the license repository. This would cause an invalid license repository ("License server host not registered in license repository").

What open ports are required for the network?

The number of open ports required will depend on the type and number of servers installed.

- License server: three ports are required for one license server and the bus and hub components it requires. Port p must be entered for the license server and the other components always use the p+1 and p+2 port numbers.
- Agent: one port is required for one agent. If this is not entered, then the p+3 port is used as standard, taken from the license server port entered.
- Model server and model versions: one port is required per independently-operated model server. This can be entered when the server is started. If the port is not given, then a port is dynamically-assigned by the license server based on the components running and is increased based on the p value of the license server entered. Port assignment is always dynamic for model versions.

We recommend that you use dedicated project license servers for each Innovator version to ensure no issues when carrying out a migration for Innovator version updates.

> **Note**
>
> We recommend that you reserve a port range for Innovator and enable communication here from clients and between the servers involved. The size of port range necessary is based on the aforementioned architecture components and the number of model servers that should be active at the same time. Ports which are no longer used are not immediately available again, depending on the operating system used. This means that you should calculate a buffer when starting a new model server.

> **Note**
>
> The ports used must be a value between 1024 and 64000.

Client Computer

How are the application programs distributed?

Innovator supports automatic software distribution under Windows. You can distribute the application programs in MSI or MSIX format via your desktop management system. If you do not have an automated software distribution, then you can use the app installer to centrally manage the application program.

Can you use a desktop virtualization solution?

Yes, you can use Innovator with popular virtualization solutions such as Citrix or Microsoft Remote Desktop Services.

You can use the Windows virtual desktop via App Attach when using MSIX installation packages.

Which application programs should be provided for users?

Depending on the task, you can provide users with suitable application programs. The model editor is a good idea for all users.

The configuration editor is required by administrative client applications for users who want to actively design the model configuration and need to edit or create a new configuration profile. The administration program is a good idea for users who are familiar with preparing models and maintaining the entire Innovator environment.

Constraints when Using Multiple Agents

The agent component is a powerful tool for dynamic administration of your Innovator infrastructure. You should avoid the following constellations when using multiple agents on one computer or when setting-up common access on a network drive.

- **Agents with Different Project License Servers and Common Project Directory**

  Multiple agents are run with different project license servers of the same version but with a common project directory.

  You cannot determine which agent is responsible for the model versions.

  You cannot create model versions or these are misassigned.

- **Agents and model servers with different project license servers and common project directory**

  Agent and model server are run with different project license servers of the same version but with a common project directory.

  Agent and model server are run on different buses and do not come into contact with one another.

  Model version states cannot be identified as unique. This may lead to mistakes being made.

Installation Examples

Installation examples show you typical scenarios for preparing and distributing Innovator components.

Client/Single Server

Description

This installation option is the default option for a client-server installation. You can use this option when starting out with your modeling team and if you do not (yet) have increased demands regarding structuring or scaling.

Set-Up

- Server Computer
    - Select a server machine with the **Server** installation type
    - Set up an open port on the server machine
    - If required, customize directories for INOPRJ and INOLIC (local hard drive)
    - Set up components as services (license server, bus, agent)
- Client computer 1-n
    - Select one or more client machines with the **Client** installation type

Graphic Overview

The following graphic shows you the different installation options.
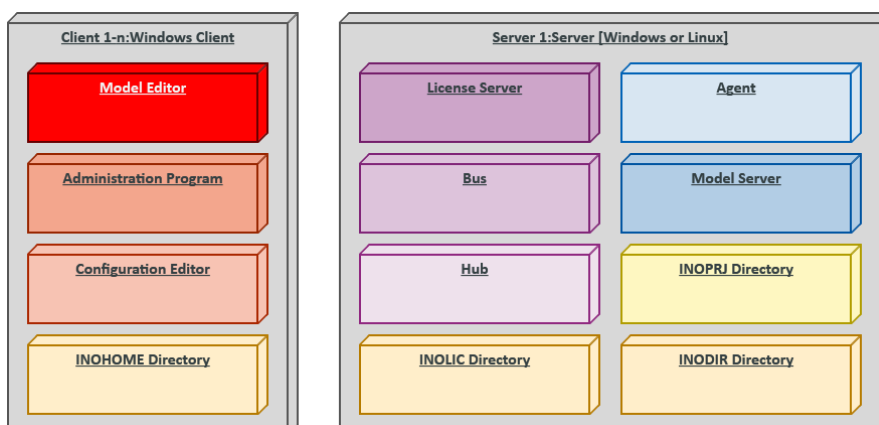
Architecture Clients and Single Server

Client/Multiserver

Description

Innovator allows you to meet your requirements should you wish to use multiple computers in a larger environment as Innovator servers.

You can use this option for working in Innovator project directories with various computers on the network at the same time and achieve an improved load distribution and resource selection.

The following example shows you the option with a project directory split on the network. You can take this example and make it work with the other options too.

Graphic Overview

Client 1-n:Windows Client	Server L:Server [Windows or Linux]	Server M1:Server [Windows or Linux]
Model Editor	License Server	Agent
Administration Program	Bus	Model Server
Configuration Editor	Hub	
INOHOME Directory	INOLIC Directory	

Server M2:Server [Windows or Linux]
Agent
Model Server

//Company:Fileshare
INOPRJ Directory
INODIR Directory

Architecture Clients, License Server and Multiple Model Servers

Set-Up

- License server computer
  - Select **Server** installation type
  - Set up bus
  - Set up an open port on the server machine
  - Customize INOLIC directory if required (local hard drive)
  - Set up components as services (license server, bus)

- Model server computer 1-n
  - Select **Server** installation type
  - Set up agent using INOPRJ start parameter on network file share
  - Set up an open port on the server machine
  - Customize INOPRJ directory if required (local hard drive)
  - Set up components as services (agent)
- Set up file share for INOPRJ and INODIR directories in the network
- Client computer 1-n
  - Select one or more client machines with the **Client** installation type
  - Set common INOPRJ project directory
  - Set common INODIR workgroup directory

Single User

Description

Innovator works best in an environment with many users. Single user operation and installation on one computer is still possible.

Set-Up

- Select **Complete** installation type
- Set-up options
  - Set up components as services (license server, bus, agent)

    Recommended for intensive use and if sufficient system resources exist for permanent operation
  - Alternatively, manually start license server; bus and agent start automatically

    Recommended for occasional use or with missing system resources for permanent operation

Graphic Overview
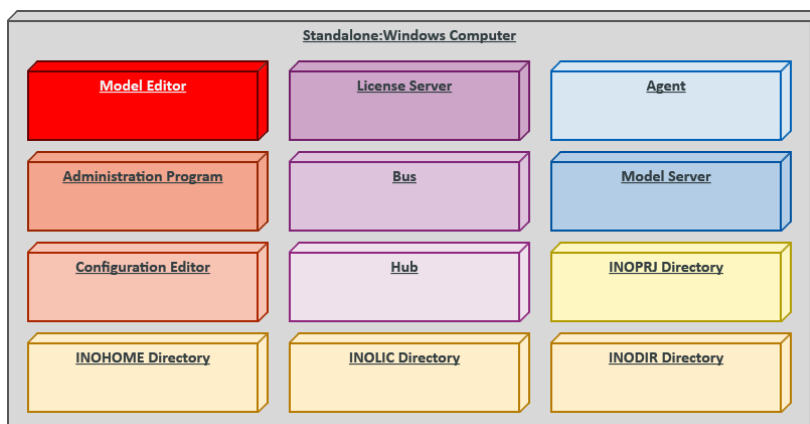
Single User Architecture

Project License Server for Innovator Versions

Description

Innovator allows you to meet your requirements should you wish to use multiple computers in a larger environment as Innovator servers and make isolated license pools available for individual projects.

You can combine this installation option with all other installation examples.

You may need to run Innovator in various versions at the same time. For example, if you still wish to access data from a finished project but do not wish to spend the time and effort migrating the project. Running the versions in parallel may be helpful when updating to the newest Innovator version to reduce migration risks and allow you flexibility without time constraints when testing and implementing migration.

The last option described here is the example used for the following set-up description and the diagram.

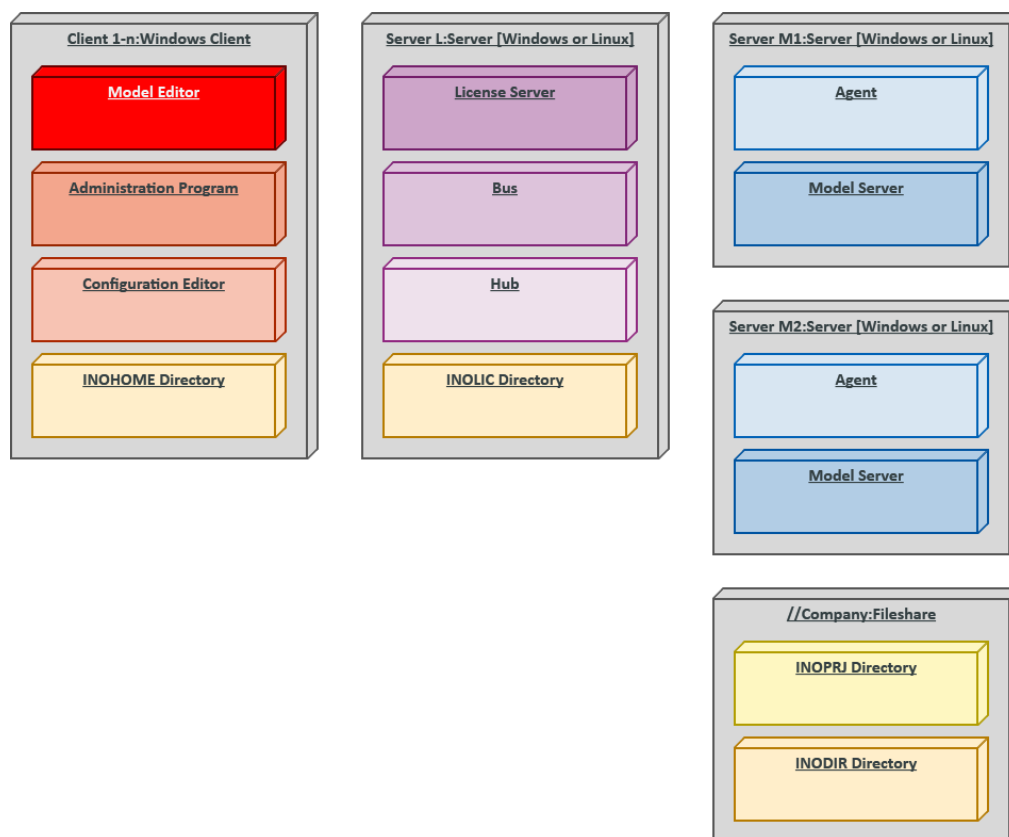
Restriction

Please refer to your signed declaration when working on upgrade licenses in parallel.

By signing the document, the license holder must

⊘
- Only install upgrade licenses in parallel with the release currently being used only during the migration period and ensuring that the upgrade works
- Ensure that the maximum number of license products assigned for the current valid Innovator license files are not exceeded at any point.

The license holder understands that violations of these points will have legal consequences.

Set-Up

- Main license server computer
  - Select **Server** installation type
  - Set up bus
  - Set up an open port on the server machine
  - Customize INOLIC directory if required (local hard drive)
  - Set up components as services (license server, bus)
  - Separate project licenses for project license server
- Project license server computer for current version
  - Select **Server** installation type
  - Set up bus
  - Set up an open port on the server machine
  - Set up components as services (license server, bus)
- Model server computer 1-n
  - Select **Server** installation type
  - Set up agent using INOPRJ start parameter on network file share
  - Set up an open port on the server machine

- Customize INOPRJ directory if required (local hard drive)
- Set up components as services (agent)
- Set up file share for INOPRJ and INODIR directories in the network for Innovator computers
- Client computer 1-n
  - Select one or more client machines with the **Client** installation type
  - Set common INOPRJ project directory
  - Set common INODIR workgroup directory

Graphic Overview

Client 1-n:Windows Client
Model Editor
Administration Program
Configuration Editor
INOHOME Directory

Server L:Server [Windows or Linux]
License Server
Bus
Hub
INOLIC Directory

Server M1:Server [Windows or Linux]
Agent
Model Server

Server M2:Server [Windows or Linux]
Agent
Model Server

Server P:Server [Windows or Linux]
Project License Server
Bus
Hub

//Company:Fileshare
INOPRJ Directory
INODIR Directory

Architecture Clients, License Servers, Various Model Servers and (Multiple) Project License Servers

Installation for Windows Systems

There are various ways to install Innovator under Windows.

Installation Files

The installation files for direct or administrative installation of the client and server are available in German and English. Language selection determines the language for installation dialogs and the default interface language setting. You can use the corresponding environment variable to change the interface language at a later stage.

Installation Files for Local Installation

Use one of the following options for direct installation on a local computer for the current user or for all users.

A. Installation File `InnovatorEnterprise-en-US.exe`

    The program checks the software installation requirements for the client and will also provide these if you have Internet access; if not, the installation process will proceed anyway.

B. `InnovatorEnterprise.msi` Installation File

    If you do not want to check and carry out software installation requirements – because you have no Internet access on the server, for example – you can use Windows Installer with the `InnovatoratorEnterprise.msi` MSI file in the `InnovatorEnterprise.zip` installation package.

Installation Files for Software Distribution with the Company

A. Windows Installer MSIX

    Load the MSIX installation package for Innovator client application and plug-ins if you wish to use the latest application supply procedures MSIX under Windows 10 or under Windows Virtual Desktop. You can distribute these installation packages using your regular desktop management system or a local app installer file which your users install. You can integrate the MSIX installation packages using App Attach on your Windows Virtual Desktop.

    If necessary, please contact our Hotline when using these packages.

B. Windows Installer MSI

    Call the `msiexec.exe` Windows Installer using the appropriate parameters for administrative installation using MSIX; you can then prepare customized installation files for installation from network.

Locally Installing Innovator

You can install Innovator clients and servers using an installation file.

Starting the Installation

You can download the installation file for Innovator from the MID website after registering either as a potential buyer or as a customer.

The following paragraph describes the installation for the supported Microsoft Windows operating system.

To start the installation of Innovator, download the `InnovatorEnterprise-en-US.exe` installation file from the customer area of the MID website and start it.

Installation Wizard

The installation file sets up the Innovator installation. The wizard checks whether Visual C++ Redistributable Packages for Visual Studio 2015-2019 (64 is installed and installs it from the internet if it is not already installed. Internet access is required for this.

The default installation paths are the same as the standard Windows paths.

- Programmverzeichnis: %PROGRAMFILES%\Innovator\15.1

  (installation destination)

- License directory: %PUBLIC%\Documents\Innovator\15.1\inolic

  Corresponds to the INOLIC environment variable

- Project directory: %PUBLIC%\Documents\Innovator\15.1\inoprj

  Corresponds to the INOPRJ environment variable

- Settings: %PROGRAMDATA%\Innovator\15.1 (All Users)

  Corresponds to the INODIR environment variable

- User-specific settings: %APPDATA%\Innovator\15.1

  Corresponds to the INOHOME environment variable

Space Requirement

The **Computing space requirement** dialog appears, if somewhat briefly. This analyses the installation files and certain prerequisites.

General License Agreement

You need to agree with MID GmbH's license agreement in the **License Agreement** dialog to be able to continue with the installation.
Read the MID GmbH license agreement and activate the check box to agree with it. Finally, click on **Next**.

Selecting Setup Type

Specify the installation directory and scope of installation with all the product functions you wish to be able to use in the **Choose Setup Type** dialog.

This dialog suggests an installation directory; this is the Windows default directory for programs `%Pro-gramFiles%\Innovator\15.1` for a first time installation.

To change the target directory, click on the **Browse…** button and set the installation directory for the program files in the **Change destination folder** dialog.

To choose a predefined installation scenario, click on the **Complete**, **Typical Client** or **Typical Server** buttons.

- **Complete** starts the installation of all Innovator components, except for the Offline Help to the selected directory. This gives you access to practically all program functions.

  This type of installation is suitable for single user installation and evaluation.

  If you do not have online access, then please install the Offline Help separately so you can access the context-sensitive help.

- **Client** installs all client components. You can use is for editing Innovator models. Tools for environment settings and configuration are also installed but server components are not.

  This type of installation is only suitable in conjunction with a server installation in the network.

- **Server** installs all components required for a server computer.

  This type of installation is suitable for a server installation in the network.

- **Custom** gives you the option of choosing which product features to install and define all the installation parameters yourself. The default settings correspond to the **Complete** installation type.

  The **Environment Settings** and **Product Features** dialogs only appear if you choose the **Custom** installation type.

Environment Settings

The **Environment Settings** dialog only appears if you chose the **Custom** installation type.

License Server

A license server manages the Innovator licensing. In order to communicate with this license server, each Innovator application needs the following information:

- The address or name of the computer that the license server is running on
- Port number used for communication between server and local computer

> **Note**
> The port number must be a value between 1024 and 64000

The information must always be added in the form `<License server host name>.<Port number>`, i.e. the different parts of information are separated by a point.

- Single User Operation

  If Innovator is to be operated without a network, then the address of the local computer (`localhost 127.0.0.1`) must be confirmed as the name in this dialog. Otherwise, the local computer name may cause problems.

  You must specify a port number within the given range.

- Network Operation with TCP/IP

  Using TCP/IP, computers are identified using both an address and a name. The name of the computer where the license server is to be run or is already running, as configured under TCP/IP, must be entered as the license server name.

  A port number that is not being used by other programs must be entered after the dot. Available numbers are usually values greater than 5000.

The license server entry corresponds to the `INOHOST` environment variable.

Shared Workgroup Directory

Innovator stores settings and certain files (e.g. parameter files and Java files for engineering actions) in a **local directory** `%PROGRAMDATA%\Innovator\15.1` as standard.

Innovator also offers the alternative option of using a **shared network directory** to carry out central specifications and to simplify updates, since they can be carried out centrally.

> **Note**
>
> All users must have read rights for the **workgroup directory**. The user making the changes to the settings in the parameter files etc. must have write rights in the workgroup directory. All users of a workgroup directory need to be in the same network and use the same license server.

Network type information corresponds to the INODIR environment variables.

The `\Innovator\15.1\` path is created below the Windows environment variables `%APPDATA%` by default for user-specific settings. This path corresponds to the INOHOME environment variable.

Directories for Model Repository and License Repository

Innovator stores model repositories in the `%PUBLIC%\Documents\Innovator\15.1\inoprj` **local directory** as default. The project directory entry corresponds to the INOPRJ environment variable.

Innovator stores the license repository in the `%PUBLIC%\Documents\Innovator\15.1\inolic` **local directory** as default. The license directory entry corresponds to the INOLIC environment variable.

You can select other locations as default for the model and license repository.

Type of Installation

With the type of installation, you specify whether Innovator is to be installed for the **current user** or for **all users** of this computer. The entries for the Innovator program group and Windows registration are then stored in the respective directory.

How to proceed

1. Only change the name of the license server computer if the license server shouldn't run on the computer which you are about to install Innovator on.

2. If you want to specify a shared working directory in the network, activate the **Use a shared working directory** check box and enter the corresponding network path.

    These files are then not installed locally.

3. To set whether Innovator should be available for the current or all users of the machine, activate the respective check box.

4. Confirm the **Environment Settings** dialog with **Next**.

Product Features

The **Product Features** dialog only appears if you chose the **Custom** installation type.

The structure of the Innovator program features appears in this dialog for you to select. For each program features you can specify whether

- it is installed on the local hard drive
- it is not installed if it is not needed
- it is not to be installed.

Innovator has the following product functions:

- **Innovator**: General DLLs
    - **Editions**: Model editor, impact analysis, beamer, DMN import/export, Excel import/export plug-ins

- **Innovator for Enterprise Architects**: ArchiMate im/export plug-in
- **Innovator for Business Analysts**: Activity-based costing, BPMN import/export and text-to-element plug-ins
- **Innovator for Information Architects**: BI, diagram layout, dimension maintenance, dimension and hierarchy export, and input/output specification plug-ins
- **Administrative clients** (administrative tools)
  - **Administration**: Administration program tool
  - **Configuration editor**: Configuration editor tool
- **Servers** (license and model servers)
  - **Model servers**: Model servers with model templates in INOPRJ directory and agent
  - **License server**: License server with bus and hub
- **Migration**: Transformer for data transfer from previous Innovator versions

To change the installation target folder, click on the topmost **Innovator** node, click on **Browse…**, and set the installation directory in the **Change destination folder** dialog.

Completion Message

The **Completion** dialog informs you once installation is complete. You can define whether you want to display the **ReadMe file** after the installation and whether you want to start Innovator immediately. The ReadMe file contains important system information about using Innovator.

> **Note**
> Before you can work with an unrestricted version, you first need to activate the Innovator products you have bought by loading a license file.

> **Note**
> To use services with Windows, further actions must still be carried out and certain precautions must be taken.

Administrative Network Installation (MSI)

> You can find all necessary information for installing clients and servers in the network here.

Administrative Installation

Purpose

If you select administrative installation, you can carry out additional installations on other computers with the same settings via the network using this installation.

The advantage of administrative installation is that as system administrator you can already decide on certain specifications (INOHOST, INODIR) from the outset. This means that even those people who might normally struggle when having to carry out installations themselves will be able to install Innovator or you can even prepare it in such a way that it will be an automated process.

Prerequisites

You require the MSI file `InnovatorEnterprise.msi` from the installation package `Innov-atorEnterprise.zip`.

Windows Installer

You can find information about the essential parameters of the Windows Installer (`msiexec.exe`) by calling:

```
msiexec.exe /help
```

You can find further information about the Windows Installer and its properties (*Property*) under:

- https://docs.microsoft.com/windows/win32/msi/administrative-installation
- https://docs.microsoft.com/windows/win32/msi/windows-installer-portal

For administrative installation, start Windows Installer with the following parameters:

```
msiexec /a InnovatorEnterprise.msi /qf TARGETDIR=<target> [ADDLOCAL=<features>]
INOHOST<License server host name>.<Port number>
```

- `/a <Product.msi>`

  Prepares the administrative installation of a product by preparing the msi file in the target drive.
- `/q[n|b|r|f]`

  Sets the user interface for administrative installation.
  - `n` - No user interface
  - `b` - Simple user interface
  - `r` - Reduced user interface
  - `f` - Full user interface (standard, recommended)
- `TARGETDIR=<target path>`

  Sets the target directory for preparing the installation files in the network. You require write rights and sufficient storage space. If you do not enter these parameters, then the first available network directory with write rights and sufficient storage space found is used.
- `ADDLOCAL=<features>`

  The value of the optional ADDLOCAL property is a features list separated by commas. It is installed locally if you want to constrain the features to be installed.
- `INOHOST<License server host name>.<Port number>`

  Example for entering a variable that should be replaced or that is not set in the MSI file.

Innovator Features

Innovator **features** and their components for the Windows Installer are listed below:

- **Innovator**: General DLLs
  - **Editions**: Model editor, Plug-ins Beamer, DMN import/export, Excel import/export
    - **I4EA**: Plug-in ArchiMate
    - **I4BA**: Plug-ins Activity-based costing, BPMN import/export and Text-to-element

- **I4IA**: Plug-ins BI, diagram layout, dimension maintenance, dimension/hierarchy export, and input/output specification
- **Admin** (administrative tools)
  - **InnoAdministration**: Administration Program Tool
  - **InoConf**: Configuration Editor Tool
- **Servers** (license and model servers)
  - **InoSrv**: Model servers with model templates in INOPRJ directory and agent
  - **InoLSrv**: License server with bus and hub
- **Transformation**: Transformer for data transfer from previous Innovator versions

Subordinate features include the direct components of the respective superordinate features.

Preparing Installation Files

When opening with full user interface, dialogs appear as described in <u>"Locally Installing Innovator"</u>. Enter the specifications you want for the installation. It is also possible to enter the specifications using the appropriate parameters.

To constrain the features to be installed, select the features to be installed.

Select a network directory as target directory that you have write rights for. Use this directory for preparing Innovator for installation of this network directory.

> **Note**
>
> If you want to use a shared workgroup directory, enter a suitable position in the network with `INODIR=<target path>`.
>
> If you want to use a common project directory, enter a suitable position in the network using `INOPRJ=<target path>`.

Once the administrative installation is complete, the directory will contain both a customized copy of the `InnovatorEnterprise.msi` file with your specifications and all files necessary for subsequent distribution.

Examples

For silent installation of an Innovator for Business Analysts client on the first suitable network directory found, the Windows Installer's command line should contain e.g.:

```
msiexec /a InnovatorEnterprise.msi /qn ADDLOCAL=I4BA
```

For interactive administrative installation of an Innovator for Information Architects client on a network directory, the Windows Installer's command line should contain e.g.:

```
msiexec /a InnovatorEnterprise.msi /qf TARGETDIR=\\srv1\Install ADDLOCAL=I4IA
```

Installing Innovator on the Network

If you have an adapted installation file for Innovator in the network, then you can install Innovator from the network on to another computer. This requires less user settings as they have already been made upon administrative installation.

> **Note**
>
> If you want to enable direct installation from the administrative installation's target directory, then you need to copy the `setup.exe` file into this directory. This is because it checks the installation prerequisites on the local computer and this is not automatically provided in the target path.

How to proceed

To install Innovator on the network:

* Navigate from your computer on the network to the computer and directory where the Innovator installation file was prepared (target path for the administrative installation).
* If you want to carry out the installation with the full user interface, then call the customized installation file `InnovatorEnterprise.msi` in this directory with `setup.exe` and follow the dialogs as described in "Locally Installing Innovator".
* If you want to run the installation in the background without the need for user interaction or with reduced user interface, then use Windows Installer's default parameters `/quiet` or `/qn`.

Company Software Deployment with MSIX

Microsoft's MSIX provides a package format for Windows apps. This combines the best features from MSIX and other installation and start techniques for applications to enable a modern and reliable package creation for preparing software within a company.

MSIX Core Concepts

What is MSIX?

MSIX (https://docs.microsoft.com/windows/msix/) is a new installation format from Microsoft, optimized to provide software within a company. MSIX has many advantages over other types of installation.

MSIX is available for installation on the following Windows versions and for Windows virtual desktop. Installation is possible on Windows 8 when in compatibility mode.

* Windows 10, Version 1709 and higher
* Windows Server 2019 LTSC and higher
* Windows Enterprise 2019 LTSC and higher

Security and Integrity Thanks to Seamless Signature Chain

All MSIX packages must be signed using a certificate valid within the company. This may be a manufacturer's certificate (as provided by MID within the scope of delivery) trusted by a global root certificate authority. A company can also (re) sign packages using certificates provided by their own internal CA. MSIX packages without a valid certificate cannot be installed.

Reliability and No Side Effects Using Virtualization

Unlike with other installation programs, installation on client computers is not implemented by MSIX using a flow which can be replicated repeatedly. The MSIX package already contains the end state of the file system and registration for the application and is then merely connected to the client computer in a virtual container. This significantly improves the quality and reliability of the installation.

Protection of Data Carriers and Network Resources Thanks to Block Optimized Transfer

Network installation and installation on a local data carrier are both managed by the operating system and are carried out in (data) block levels. This enables e.g. only precisely the amount of data required that has actually changed between the two versions to be deployed via the network when updating the application. This is particularly advantageous when you have a large number of client computers and can save unnecessary network traffic.

Customizability and Extendability Using Modification Packages

MSIX enables companies maximum control over which content is installed on their client computers. This means that every application package can be modified using modification packages. Modification packages can be created by a software provider or also by the customer themselves. This enables a customer to easily customize installation of software.

Planning Installation

MSIX installation packages provided by Innovator only contain the application programs, as only these need to be deployed on a large number of client computers. You should use the EXE or MSI installation program to install the server components.

You should first set your required installation options and customer-specific environment settings.

Also download "Client Programs for Windows 10 (MSIX)" from the customer area and unzip it in a local directory.

* Main Package

   The `MID.Innovator_15.1.nnnnn.0_x64.msix` application package contains the Innovator appliction programs.

* Plug-In Modification Packages

   Packages called `MID.Innovator.Plugins.<Pluginname>_15.1.nnnnn.0_x64.msix` contain the various plug-ins included within scope of delivery. Our support would be happy to provide you with MSIX packages for any other on-demand or customer-specific plug-ins which you previously obtained from MID.

* Offline Help Modification Package

   The `MID.Innovator.Help_15.1.nnnnn.0.msix` package installs the offline help.

* Modification Packages for Environment Settings

   The `MID.Innovator.inoenv` and `MID.Innovator.Settings.msix` modification packages contain templates for customer-specific environment settings (e.g. INOHOST). You need to change the settings before installation.

Select the packages relevant to your environment.

Making Customer-Specific Settings

You must set at least one basic environment variable on the client computers so that the Innovator application program can contact the servers.

There are two options for this.

A. Traditional deployment of registration keys required (e.g. Desktop management, active directory group policies)

A program installed using MSIX can access all registration entries available to the user on the computer. This means that Innovator is able to access the Innovator environment variables even if they are deployed on client computers separate to the MSIX package.

Use this option if you already have an established process for this and do not wish to carry out further customer-specific customizations.

B. Packing Settings in a Customer-Specific Modification Package

If you wish to install the Innovator environment fully using MSIX or wish to deploy additional files for Innovator onto client computers (e.g. Your own engineering actions, icons, documentation templates etc.), then we recommend that you create your own modification package.

MID would be happy to tailor MSIX installation packages to suit your plug-ins. Please contact our support for this.

> **Note**
>
> It is not currently possible to control installation of single applications within an MSIX package. This is why the model editor, configuration editor and administration program are all installed via MSIX and can be accessed via the model editor.
>
> You can prevent this by adding an entry in the `msix.ini` file via a modification package.
>
> Add the following content to `msxi.ini` if you which to prevent the administration program and configuration editor from starrting:
>
> ```
> [Allow]
>   Administration Program = 0
>   Configuration Editor = 0
> ```

Application Deployment

Installation on the client computer does not require administration rights thanks to the digitally signed MSIX packages managed by the operating system. Each software package is individually installed in the user context for each user. Multiple users on a client computer only need install the application once.

There are two options for deployment of the application to client computers.

A. **Software Deployment Using Client Management System**

MSIX packages can be deployed by all client management systems, such as SCCM. Use this option if you already have established structures and processes within your company. You and your client management team will profit from the low test outlay, use of few resources and quicker runtimes.

See MSIX App Distribution for more information.

B. **Individual Software Distribution Using the App Installer**

You can make Innovator available via the App Installer if you do not use a client management system or are able to use a degree of freedom within your company when installing Innovator.

See App Installer File Overview for more information.

This method enables individual management using both application owners and centralized deployment, including automatic updates.

An App Installer file provided individually for the customer defines the scope of installation (main and modification packages) for Innovator, including the central storage location of the MSIX files in the network. The defined packages are installed upon installation of the App Installer file on the client computer. It is easy to update the updated MSIX packages to a central location when updates

are available for Innovator or individual plug-ins. The client computer can then automatically install the updates (few resources used thanks to block-optimized transfer).

Creating MSIX Modification Packages

You can use an MSIX modification package to prepare files or settings which are not contained in the standard MID packages for your Innovator installation.

Preliminary Considerations

When should you create an MSI modification package?

You should create an MSI modification package for an application package f you want to prepare files or settings which are not contained in the standard MID packages for your Innovator installation.

Which information can be deployed using an MSIX modification package?

You can deploy all types of file types and settings in the Windows registry using an MSIX modification package.

The following constraint applies for files: files can only be deployed

- under %PROGRAMFILES%\Innovator\… (e.g. Plug-ins) in the Innovator program directory or
- under %PROGRAMDATA%\Innovator\… (e.g. Java Engineering actions, icons) in the directory typically used as INODIR.

User-specific files cannot be deployed in this way (storage location for INOHOME is typically %APPDATA%\Innovator\...).

The following constraint applies for registration information: you can enter any content for registration under Local Machine (HKLM) and Current User (HKCU). This registration information is only visible for Innovator (and users in the Innovator context); no other programs can see it.

> **Note**
>
> If you already have an Innovator registration key in your Windows system, then an Innovator version installed using MSIX can also access this. If the same registration key is installed again with an MSIX package, then these keys take precedence over the existing keys in the system.

What do you need to create a modification package?

You require software which enables you to create an MSIX modification package. There are various third party products which can be used for this. MID recommends the free Open Source toolkit MSIX Hero.

The MSIX Packaging Tool provided by Microsoft is also useful for this.

You require local administrator rights to be able to use these programs.

All MSIX packages must be digitally signed. You either require a local signature certificate or you need to be able to sign a package using a signature position in your company to be able to deploy within your company.

Creating a Modification Package - Example

Context

The following example shows you how to create a modification package for your own Innovator plug-in. This is done using MSIX Hero software.

How to proceed

1. Create an empty directory where you which you structure the modification package, e.g. `C:\Temp\MyPlugin`.
2. Start the MSIX Hero program and select **Tools>Modification Package Generator**.

   The **Create modification package** dialog appears.
3. Enter a meaningful name for the package content as **Displayed package name** in the **Properties** tab.

   This name is shown during installation and within the system.
4. Enter your company's name as **Displayed publisher name**.

   This name is shown during installation and within the system.
5. Enter a name which makes this modification package clearly recognizable on client computers under **Package name**.

   MID recommends the nomenclature `MID.Innovator.15.1.Plugins.<MyPlugin>` for Innovator plug-ins. This enables you to manage versions of your plug-ins for various Innovator versions.
6. Enter the name that corresponds to the name on the certificate in **Publisher name**; this is the name you will later use to sign the package, e.g. CN=CompanyName.
7. Enter a meaning version number for your plug-in under **Package version**.
8. Select the MSIX package for the Innovator main program under **Parent package**.
9. Click on **Create modification package...** and select the directory you created above.
10. Jump to this directory in the Windows explorer and create a sub directory structure `VFS\ProgramFilesX64\Innovator\Plugins`.
11. Copy your plug-in's directory into this folder.
12. Select **Tools>Pack folder to MSIX** in the MSIX Hero program.

    The **Pack MSIX package** dialog appears.
13. Enter the directory you created above for the modification package in the **Source + target** tab.

    You can also select another output directory for the file to be created if you require.
14. If you have a signature certificate that is installed locally, then select the **Sign this package** option.

    The **Signature** tab appears. Select the certificate here.
15. Click below on **Pack folder to MSIX package**.

    A MSIX file is created. If you have added a signature to it yourself, then you can get going with installing it on a computer which has the Innovator main package on straightaway. Your plug-in is then available once you reboot.

Setting-Up App Installer Distribution

You can use Microsoft's App Installer to install and update application packages via a website or company-internal network.

What is App Installer?

App Installer is a plug-in for Windows 10 created by Microsoft; you can use it to install application packages via a website or company-internal network and update them automatically.

Mechanisms which provide a similar function to the Windows store are used. A company can use a private application package if they find the Windows Store unsuitable. No extra software products are required for the desktop management (e.g. SCCM).

You can find basic information and the technical documentation here: https://-
docs.microsoft.com/windows/msix/app-installer/app-installer-file-overview

Preliminary Considerations

When does it make sense to use App Installer for distribution?

It makes sense to use the App Installer if you wish to frequently distribute new versions of client soft-
ware or plug-ins for Innovator (including bug fixes) to users and make the administrative and operative
side of things as easy as possible.

What do you require for installation with App Installer?

You require the MSIX application packages, network share or a webspace which you have write rights
for and the user read rights.

The Microsoft App Installer must be installed on Windows 10 computers (https://www.-
microsoft.com/p/app-installer/9nblggh4nns1). App loading must also be activated (sideloading). This is
activated as standard from Windows Version 2004, OS Build 19041.789.

Creating an App Installer Installation File – Example

You only require a simple text editor when creating an App Installer installation file. Our support can
provide you with a template and support when creating the installation file.

The following `innovator.appinstaller` file's content describes Innovator installation with the BPMN
plug-in package:

```xml
<?xml version="1.0" encoding="utf-8"?>
<AppInstaller
   Uri="file://[individual network path]/innovator.appinstaller"
   Version="1.0.0.0"
   xmlns="http://schemas.microsoft.com/appx/appinstaller/2018">
   <MainPackage
      Name="MID.Innovator.15.1"
      Version="15.1.1.20127.0"
      Publisher="CN=MID GmbH, O=MID GmbH, STREET=Kressengartenstrasse 10, L=Nure-
emberg,
      S=Bavaria, PostalCode=90402, C=DE"
      Uri="file://[individual network path]/MID.Innovator_15.1.1.20127.0_x64.m-
six"
      ProcessorArchitecture="x64" />
   <OptionalPackages>
      <Package
         Name="MID.Innovator.15.1.Plugins.BPMN"
         Version="15.1.1.20127.0"
         Publisher="CN=MID GmbH, O=MID GmbH, STREET=Kressengartenstrasse 10,
L=Nuremberg,
         S=Bavaria, PostalCode=90402, C=DE"
         Uri="file://[individual network path]/MID.Innovator.Plugins.BPMN_
15.1.1.20127.0_x64.msix"
         ProcessorArchitecture="x64" />
   </OptionalPackages>
   <UpdateSettings>
      <OnLaunch
         HoursBetweenUpdateChecks="12"
```

```
        ShowPrompt="true"
        UpdateBlocksActivation="true"/>
    </UpdateSettings>
</AppInstaller>
```

The following settings are relevant:

- The URI attribute describes the company-internal location of the installation files; the user can access this at the point of installation. This must never be altered for App Installer information. The Innovator MSIX packages can be referenced in the respective individual version.
- The `AppInstaller`'s element's `Version` attribute describes your company-internal version of your App Installer file.
- The `MainPackage` and `Package` elements' `Version` attribute must be the same as the referenced installation package.
- Add a `Package` entry under `OptionalPackage` for all modification packages respectively.
- You can use the `UpdateSettings` section to determine how updates for the package should be treated. You can find documentation about this under https://-docs.microsoft.com/uwp/schemas/appinstallerschema/element-update-settings

  The `UpdateBlocksActivation` attribute controls e.g. that a user can only start the application if the update is also installed. This ensures that a particular version is always used.

Double-click on the App Installer file to activate a local installation. This will show any existing errors.

Provide the App Installer file on the specified path in the network and inform the applicable user groups that installation can be carried out. The user must always initiate first time installation of the App Installer. Subsequent updates can then be distributed automatically.

Activating a Program Version Using App Installer – Example

An updated version of the BPMN plug-in package should be distributed for the aforementioned software installation.

Copy the new package, e.g. `MID.Innovator.Plugins.BPMN_15.1.20201.0_x64.msix` to the network next to the other packages.

Open and edit the App Installer file as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<AppInstaller
   Uri="file://[individual network path]/innovator.appinstaller"
   Version="1.1.0.0"
   xmlns="http://schemas.microsoft.com/appx/appinstaller/2018">
   <MainPackage
      Name="MID.Innovator.15.1"
      Version="15.1.20127.0"
      Publisher="CN=MID GmbH, O=MID GmbH, STREET=Kressengartenstrasse 10, L=Nure-
emberg,
      S=Bavaria, PostalCode=90402, C=DE"
      Uri="file://[individual network path]/MID.Innovator_15.1.20127.0_x64.msix"
      ProcessorArchitecture="x64" />
   <OptionalPackages>
      <Package
         Name="MID.Innovator.15.1.Plugins.BPMN"
         Version="15.1.20201.0"
         Publisher="CN=MID GmbH, O=MID GmbH, STREET=Kressengartenstrasse 10,
```

```
L=Nuremberg,
        S=Bavaria, PostalCode=90402, C=DE"
        Uri="file://[individual network path]/MID.Innovator.Plugins.BPMN_
15.1.20201.0_x64.msix"
        ProcessorArchitecture="x64" />
    </OptionalPackages>
    <UpdateSettings>
       <OnLaunch
        HoursBetweenUpdateChecks="12"
        ShowPrompt="true"
        UpdateBlocksActivation="true"/>
    </UpdateSettings>
</AppInstaller>
```

Modify the `Version` attribute for the App Installer file. Change the reference to the BPMN Plug-in package, so that it is inline with the new version (`Version` and `URI` attributes).

Save the App Installer file.

All Innovator packages installed using the App Installer automatically check whether a new BPMN plug-in package exists, install it and start Innovator along with the updated plug-in.

Changing the Installation and Environment Variables

As is the case with Windows, you can change the Innovator installation in the system at a later stage. You can change environment variables in the administration program.

Changing the Installation at a Later Stage

The **Programs and Features** function can be used in the system control for changing entries made upon installation at a later stage - as is often the case in Windows.

You have the following options for an existing installation:

- **Repair** should be required only rarely since this replaces the installation files.
- **Change** can be used after an incorrect installation to correct or expand the installation, e.g. from a single client installation to a complete installation.
- **Uninstall** deletes Innovator from the computer. Modified files such as your license file, repositories that you have created, configuration files for the client in the home directory and temp directories are retained.

  User-specified settings can also be removed during interactive removal of Innovator. Click on **Change** in the control panel to do this.

> **Note**
>
> If you use the `InnovatorEnterprise-en-US.exe` bundle installer for installation, then no modification option appears in the interface for **Modify**.
>
> You must use the `InnovatorEnterprise.msi` MSI file for this; this was stored in the *C:\ProgramData\Package Cache*d irectory on your system upon installation.
>
> It can be found here in a sub directory with the same name as the UUID and the installation version number (e.g. *{4561E94B-D563-44A9-A713-E2FD8CE0081D}v15.1.20225*). Search for "v15.1" or for the MIS file in the `Package Cache` directory and use the newest version.
>
> Double-click on the MSI file to see the options described above.

Changing the Environment Variables

To change the values of the Innovator environment, use the **Environment** backstage view in the
 Administration Program or the **Options** dialog.

The current environment variables are displayed and managed separately for the system and the user. User settings have priority over system settings.

> **Note**
>
> System variables can only be changed, created or copied from the user variables if you start the program as a Windows administrator.

> **Note**
>
> The relevant directories are not copied when you change the environment, which means that the specifications can only be a reconstruction of a change at file level.

You can use buttons to add or delete environment variables and to copy the value of the selected environment variable to the user or system variable. (The last option requires the Administration program to be run as Windows administrator.)

Windows environment variables in the form `%EnvironmentVariable%` can also be used for directory specifications, e.g. %APPDATA% or %HOMEPATH%.

> **Note**
>
> You can also make use of UNC notation for path specifications.
> Example: \\userver\innovator\15.1\rep

The environment variables are stored in the Windows registry.

Table: Overview of Environment Variables

Meaning	Variable	Description
Interface language	INOLANG	Language in which to work with Innovator. The installation file language is used as default.
License Server Computer	INOHOST	Computer on which the license server is to run and details of the communication channels in the format <License server host name>.<port number>
Settings (parameter settings)	INODIR	Directory path that the parameter files are or will be stored on; icons and the Java files for engineering actions etc. can also be found here. You can use a shared workgroup directory and, in doing so, use common settings.
License directory	INOLIC	Directory path where license repositories are stored as standard.
Project directory	INOPRJ	Directory path where you will find or can store repositories as standard. Configuration files for the model server can be found here.

Meaning	Variable	Description
Temporary Directory	INOTMP	Optional: In Windows, Innovator stores temporary files to the directory specified in the TEMP environment variable by default. If this variable is not set, if this causes problems or if you want Innovator to use another directory for storing temporary files, set the INOTMP variable.
User-specific settings	INOHOME	Optional: The parameters in the INODIR directory are intended as central defaults. The individual user can override these specifications using their special settings. These are stored in the INOHOME directory. If this variable is not explicitly set, then a subdirectory in the %APPDATA% directory specified by the operating system is automatically used.
Custom Icons	INOICON	Optional: Innovator manages the icons used in the model editor and administration program, in icon libraries. It manages the bitmap icons used in the Configuration Editor in subdirectories in from $INODIR\icons. You can specify further directories (preferably one) as the source for certain icons with INOICON to separate icons included within the scope of delivery and your own icons. Create the icon directories locally if network access takes too long.
Directory for the Innovator Help	HELP	Optional: Storage location for help files stored locally if Internet access is not possible.
Server response interval	INOSRV_ INFORM_ TIME	Optional: If the connection between the client and the servers in distributed environments is provided by network elements that close the connection between client and server if it has been inactive for a certain period of time, then an interval can be set here, after which time the client will send a dummy telegram to the server. The value is the time in seconds. The usual intervals are 300 to 1200 seconds.
NAT Host-name	INO_NAT_ HOSTNAME	Optional: Enables environment integration with Network Address Translation (NAT). You only need to set the environment variable on computers which the license server is not running on. The external computer with the host name given as the environment variable must be released to a valid IP address via DNS.
		All components use this host name when communicating, regardless of the host name logged-in to that computer. The set NAT host name is shown as host name in the administration program and in all interfaces.
		Innovator does not support implementation of communication ports via NAT.

Meaning	Variable	Description
Network Card	INO_NAT_ IPADR	Optional: Enables to specify the network card for communication with the model server or agent for a machine with multiple network cards running a model server or agent. Must be set in the system variables. If the environment variable is to be used for an agent, then the agent must be reinstalled as a service after setting the environment variable.

> ⚠ **Attention**
>
> All Innovator programs must be shut down on the computer so that changes can be implemented. The next time an Innovator program is called, the new settings will be used.

> ⚠ **Attention**
>
> A change to the computer name in the **INOHOST** entry means that the computer which the license server is running on has changed. You should bear in mind that in this case the Innovator license will have to be re-activated by MID GmbH.

> ⊘ **Restriction**
>
> If INOPRJ is changed once an agent service has been installed on the system environment, then the agent service needs to be uninstalled and reinstalled. This is because INOPRJ is set for the agent upon service installation (inoagent does not read the registration).
>
> If INOLIC (or INODIR if INOLIC is not set) is changed once a bus service has been installed on the system environment, then the bus service needs to be uninstalled and reinstalled. This is because the log directory is set for the bus upon service installation (inobus does not read the registration).

Environment Variables for Innovator Services

You can run license and model servers as services in Windows.

Use the 🔴 **Administration Program** (`InnoAdministration.exe`) to mange services.

> **Note**
>
> The Innovator services can only be added, copied, applied, removed or changed if you start the program as a Windows administrator.

To manage the services, open the **Services** backstage view in the **Administration Program**.

You can define the following as the start type:

A. Automatically (with Windows start)

B. Automatically (delayed start) or

C. Manual

> **Note**
>
> **Automatic** start type is recommended for the license server.
>
> **Automatic (delayed start)** start type is recommended for the model server.

If problems arise, you can increase the waiting time value for the automatic start type in seconds using the environment variables.

Table: Environment Variables for Innovator Services

Meaning	Variable	Description
Waiting time upon automatic server start	INO_ SERVICE_ START_ WAIT_TIME	You can set the time an Innovator service with the **Automatic** or **Automatic (Delayed Start)** start type waits upon starting before it accesses other services. The value sets the time in seconds that an Innovator service waits upon starting before it enables other services to launch before it, e.g. "30" (seconds).

Installing an Update

There are various measures which need to be met depending on the type of Innovator update.

Composition of the Technical Version Number

The complete technical version number for an installation includes the specifications for the Major Version (V), Minor Version (M), Release (R) and 5 Digit Build Number (B). In principle, the technical version number is composed as follows: V.M.R.B (e.g.15.1.1.XXXXX).

Version

Major version and m*inor version* form one version of Innova*tor.*

Only different versions can be installed and operated side by side. In other words, you can install Innovator 15.1 alongside Innovator 15.0.1.

Installing a newer, more up-to-date version is known as an upgrade.

The Migration Manual (see browse sequence) mainly contains information about upgrades.

> **Note**
>
> Operating a new major release requires an upgrade order, which you can download from the customer area.

Release

Release and hotfix, if applicable, make up an Innovator release.

Releases **cannot** be installed side by side since they use the same items in the Windows registry.

If the build number has been changed in a release, this shows the presence of a hotfix. Hotfixes change the existing installation and are compatible with the interface.

Installation of a newer, more up-to-date release is known as an upgrade.

It is also possible for a hotfix to exist as a single program and library if required. Your support team can provide this for you. This need to be inserted manually at the right place.

To update to a new release, follow these steps:

1. Close all programs and services for the current version.
2. Back up certain existing data
3. Install the new release

Close the current version

You must make sure you have closed all programs running for the current version before installing a release. Services running are often forgotten and can be closed using the administration program.

Backing-Up Your Own Data

If you require a fallback solution, then you need to move the old installation to another folder before you install the new release or rename this folder.

> ⚠️ **Attention**
>
> If you have directly or indirectly modified files (ini, xml, css) in the $INODIR directory, save these files to a different folder. Look out for files that have a newer time stamp than the software itself.
>
> You should back-up these files so that your own settings will not be lost.

To update to a new release, i.e. to install a release or a hotfix (e.g. from version 15.1.1.XXXXX or 15.1.3.XXXXX to version 15.1.3.YYYYY) you *no longer* need to uninstall the existing software as was the case with previous versions. Set-up of the new release always automatically uninstalls the complete old release.

If you used the repository Demo, then this will be renamed upon implicit uninstallation.

Installing an Update or Upgrade

When an update is installed it will acknowledge that there is data in the installation directory. The new files will be saved alongside the existing files or will overwrite these.

Following installation, the license repository will be loaded from the license server and verified for validity.

A. If the major and minor versions are still the same upon update (e.g. 15.1.2 to 15.1.3), then the license repository is adopted and you can carry on working with Innovator without having to reactivate the license.

B. If the version has changed (i.e. an upgrade, e.g. 15.0 to 15.1), the license repository needs to be activated by the manufacturer's order processing department (auftragsbearbeitung@mid.de).

Exchanging Single Components

It is possible to switch out single files causing issues (e.g. DLLs) with corrected versions to solve problems which are normally customer-specific.

Please note the following procedure when switching the files on Windows to ensure reliable operation of Innovator:

- Inform the teams affected that the server may be temporarily offline
- Working on affected models is only possible once the server has been restarted (if the server was affected)
- Shut down the servers or services which the updated components should be added to
- Replace the server or client components
- Restart the servers or services and test to see whether the switch out was successful

Installation for Linux Systems

Only server components are on Linux systems (openSUSE, SUSE-Linux and Red Hat Enterprise).

Prerequisites

Note the information on the required hardware and software platform, in particular the restrictions to specific distributions and versions in the case of operating systems.

Innovator can only be installed on the server side on Linux systems.

Client interfaces are only available for administrative tasks for Linux.

You require the `Linux_opensuse150_64Bit.zip` installation package.

The Java Standard Edition (SE) version 11 and higher must be installed for agent and hub to work.

Installation Package

If you are a registered customer with a service agreement, then you can get the operating system-specific Innovator installation files as a ZIP file in the exclusive customer area. You can find information about the products and installation in a separate document (PDF) and in this Help.

You can either install a hotfix as a complete installation package or as single programs and libraries. If you choose the latter option, then the files need to be copied to the right location once the current version has finished running the program.

Server Component Installation

Install the server components using the installation script and set the environment variables.

Preparing the Installation Files

Installing Innovator onto a Linux system requires knowledge of Linux and should by carried out by the system administrator. Please make sure you close any programs running on the current version first.

- Log-in as the super user: login as `root` or via `su root`
- Unzip the installation package that you have loaded from our homepage into a temporary directory.

Initializing the Installation

Start the installation script `install.sh` in the temporary directory to install Innovator on the respective system.

How to proceed

1. Start the installation script:
   `sh install.sh`

   You will be asked for the root directory for the Innovator installation.

2. Specify the directory where you want Innovator to be installed, e.g. `/usr/innovator/15.1`.

   > **Note**
   > If the installation directory does not yet exist, it will be created.

   The actual installation now takes place. The files are copied to the specified directory.

Setting Environment Variables

You are now requested to set the necessary environment variables for each Innovator user. The values assigned to these variables should be recorded in scripts which are then processed when the user logs-in, for example, in the login script (`.cshrc`, `.kshrc`, `.login` depending on the shell used). This makes Innovator immediately available after you log-in.

> **Note**
>
> The following assumes that Innovator has been installed in the directory `/usr/innovator/15.1`. If you selected a different path, it must be entered accordingly.
>
> The operating system instructions are based on the `csh`. If you are using another shell, you have to modify the command accordingly.

Search Path Extension

Add the installation directory to the search path. It contains Innovator programs.

* Example: `setenv PATH $PATH:/usr/innovator/15.1`[1]
* The LD_LIBRARY_PATH must be extended under Linux if you wish to use the Java API

> **Note**
>
> For further information regarding the general architecture of Innovator and specifying computer names and directories see "Innovator Architecture".

Defining Network Settings

Licensing is controlled by a license server. Each Innovator application requires the name of the license server computer and the port number the license server can be found under to be able to use this license server. The information must be in the form `<License server host name>.<Port number>`, i.e. the different parts of information are separated by a point.

> **Note**
>
> The host name may not contain spaces or two consecutive periods.
>
> The port number must be a value between 1024 and 64000

INOHOST

Specify the identification of the license server, e.g.:

`setenv INOHOST userver.15100`

> **Note**
>
> The TCP/IP name of the computer must be specified as the host name. The port number must be an unoccupied port number. A list of used port numbers can be found, for example, in the `/etc/services` file.

---

[1]Please note that Red Hat Enterprise Linux uses "export" instead of "setenv" and a slightly modified syntax to set system variables. Example: export INOHOST= userver.15100

In the example userver.15100, userver is the name of the computer on which the license server is to run. 15100 is the port number used for communication between the server and the local computer.

INO_NAT_IPADR

If a computer running a model server or agent has multiple network cards, then the INO_NAT_IPADR environment variable can be used to specify which network card a client uses to communicate with this model server or agent. INO_NAT_IPADR must be set in the system variables.

If the environment variable is to be used for an agent, then the agent must be reinstalled as a service after setting the environment variable.

The IP address to be used is supplied to the agent via a call argument.

Example:

INO_NET_IPADR = 192.168.4.233

-Dde.mid.innovator.ipadr=192.168.4.233

Specifying File Locations

INOROOT

Enter the absolute path of the directory which you installed Innovator on:

```
setenv INOROOT /usr/innovator/15.1
```

> **Note**
>
> To operate Innovator it is not necessary to define INOROOT as an environment variable. We are using it here because it is advantageous for defining other environment variables.

INOLIC

This variable sets the directory where the license repository will be stored by default. Set the INOLIC directory, for example, by entering:

```
setenv INOLIC $INOROOT/inolic
```

INODIR

This variable defines the directory where the parameter and help files for Innovator are located. Set the INODIR directory, for example, by entering:

```
setenv INODIR $INOROOT/inodir
```

INOPRJ

This value defines which directory contains the Innovator repositories by default. Specify the absolute path name of the directory:

```
setenv INOPRJ $INOROOT/inoprj
```

The setuid bit also needs to be taken into consideration when assigning rights for the repository directory; this is because the server processes are provided with this upon installation.

Specifying User-Specific Settings

INOLANG

Define the language in which Innovator should run. If you want to work with German menus, dialogs and message texts, enter:

```
setenv INOLANG de_de
```

If you want to work with English menus, dialogs and message texts, enter:

```
setenv INOLANG en_us
```

INOTMP

Under Linux, Innovator uses the directory specified in the TMPDIR environment variable for storing temporary files by default. If this variable is not set, if this causes problems or if you want Innovator to use another directory for storing temporary files, set the INOTMP variable. Enter the directory for storing temporary files:

```
setenv INOTMP $INOROOT/tmp
```

Please note that the sticky bit needs to be set for the directory for storing temporary Innovator files as e.g. this may cause error messages for the documentation.

INOHOME

Set the path component which the user-specific settings should be stored in. If you want to change the INOHOME directory, enter e.g.:

```
setenv INOHOME $INOROOT/inohome
```

If the environment variable's value is empty, no user-specific settings are saved and used.

This completes the installation of Innovator.

Communication between Server and Client

Innovator requires Message Queues for communication between the Server and the Client (e.g. Oracle and MySQL).

The default on most operating systems is too low for message queues, particularly when a lot of clients are using it. If this is the case, an error message "cannot create message queue" often appears.

We recommend increasing the value of the message queue to at least 256.

You can verify the current value with the command `ipcs -ql` when using **Linux**.

Input the number of available message queues in the `/etc/sysctl.conf` file:

```
kernel.msgmni = 256
```

The new value will be adopted upon reboot or directly by inputting `sysctl -p` in the console.

The amount of message queues being used in the current system should be verified after a certain amount of time using the `ipcs -q` command (in Innovator or databases). Adjust the value in such a way that it offers adequate reserves.

`ipc` does not alter the message queues of processes which have been unexpectedly ended. If necessary, these can be engaged with the `ipc` command so that the message queues can be used again.

Setting Agents as Services

Agents maintain and manage all models in the version control. You can set them as services.

Prerequisites

Apache Commons Daemon is used as service manager. You need to have installed the jsvc package under Linux for this.

The minimum requirement for Java has been changed from version 8 to version 11.

You require a new service wrapper version for Java programs under Linux (jsvc); this is because the previous version 1.0.13 cannot run Java programs which were developed with Java version higher than 8.

jsvc in version 1.2.3 is included within the scope of delivery for Innovator with the Linux installation package as current jsvc versions are not included in many Linux repositories. The jsvc version provided by MID must be configured with a Java 11 version.

Setting Up InoAgent.service

`$INODIR/java/InoAgent` contains the `InoAgent.service` file. Copy this file into the `/etc/sys-temd/system/` directory.

In some cases you may need to change certain entries in this file.

- Environment=USER

  Enter a user that has environments rights in the central user administration here.

- Environment=INOHOST

  Enter the INOHOST value used here.

- Environment=INOEXE, Environment=INODIR, Environment=INOPRJ and Environment=JARHOME

  If you have not installed Innovator under `/usr/innovator/15.1`, then you need to replace this section of path with the installation directory you chose.

Start service

Use the `systemctl` command to use the service.

The start command is e.g. `systemctl start InoAgent`.

Exchanging Single Server Components

Server components can be updated.

It is possible to switch out single files causing issues with corrected versions to solve problems which are normally customer-specific.

Please note the following procedure when switching the files on Linux to ensure reliable operation of Innovator:

- Inform the teams affected that the server will be temporarily offline
- Working on affected models is only possible once the server has been restarted
- Shut down the servers or services which the updated components should be added to
- Replace server components
- Restart the servers or services and test to see whether the switch out was successful

Licensing of Innovator

You can only use the complete range of Innovator functions if you have the respective license. Licensing fee-based editions occurs either using concurrent user licenses (license repository) or named-user licenses (subscription).

License Model

You can only use the complete range of Innovator functions if you have the respective license. You can obtain a license via our distribution partners or directly from MID at vertrieb@mid.de.

Innovator supports two alternative license models:

- **Concurrent User License Model** (floating licenses always multi-user licenses for one or more workstations)

  Licenses with a license repository are managed via a main license server and can be distributed to internal and external projects. Licenses are floating licenses, i.e. not linked to a particular user and normally without time limitation.

  Licensing depends on the number of users who work simultaneously with a given Innovator product. The license only applies to the Innovator version it was issued with and for older versions. You require a current maintenance agreement to receive a license for the current Innovator version.

- **Named User License Model** (user licenses)

  Licenses are managed by an MID-operated license server (subscription server). The licenses are linked to concrete users and require registration and online access to the subscription server.

  The user can always use the most current Innovator version with their valid user license. The license is valid for the time frame in which an active license agreement exists. Once the license agreement has ended, the user license can no longer be activated by the subscription server and further use of Innovator is no longer permitted.

Innovator can also be installed as a **Professional Edition** always complete as Enterprise Modeling Suite. The license used by the registered user when accessing the program determines which products and functions are available.

Free Use of Innovator for Business Analysts (Personal Edition)

The single-user license for BPMN diagrams is installed along with the free Innovator for Business Analysts **Personal Edition** and is available immediately.

Demo License and Test License

Following the installation of Innovator in the Enterprise edition, a demo license is active from the first time the program is run. This can be used for 60 days with a restricted number of model elements. This demo license enables a single user on one computer access to all Innovator functions. You only need to start Innovator to be able to use the demo mode. All components necessary and a demo model are started automatically.

Our distribution partners can issue a test license without any limits at any time for you if you have demanding requirements when testing and evaluating the software, e.g. multi-user operation.

Starting and Closing Innovator

You can start the Innovator interface via the Innovator <Version> program group.

Starting the Innovator Interface

A license server needs to be running to be able to start the Innovator interface, i.e. the model editor. This is automatically started, if applicable (Demo and Personal edition). The license server runs centrally and manages all Innovator licenses.

> **Note**
>
> However, it is also possible to start the model editor if the license server (INOHOST) is not set, set incorrectly, or not running. In this case, a dialog appears. You can change the Innovator environment variables there. This means that you can, for example, enter a correct license server during the normal program start process.

A model server needs to be running if a model is to be edited. This is also automatically started, if applicable (Demo and Personal edition).

Start ⊕ **Innovator** in the **Innovator <Version>** program group (`innovator.exe`).

You will be asked if you want to register your Innovator version for the Innovator log the first time you start Innovator. It is also possible to register at a later stage.

Innovator Server for Licenses and Repositories

Two programs always need to be running in the background to be able to edit Innovator data; they do not require an interface as they are automatically started and stopped.

License and model servers must not be blocked by Windows if you wish to use Innovator.

- License Server (`inolsrv.exe`)

  The license server manages and monitors the licenses you purchase and ensures the cohesion of the Innovator network interconnection with network operation. The licenses are kept in a pool and only taken out when the corresponding edition needs to be used (floating licensing).

  > **Note**
  >
  > Other Innovator programs can only be started when the license server is running. If the license server runs centrally due to a network installation, the local starting of the license server is eliminated.
  >
  > The Personal edition is a single user edition and automatically runs the license server.

- Model Server (`inosrv.exe`)

  Models servers manage the model data you work with. They ensure data consistency and prevent different users simultaneous accessing elements at the same time. Before a model can be worked on, the corresponding model server must be started.

  > **Note**
  >
  > The Personal edition is a single user version and automatically runs the model server.

Closing Innovator

Select the **Model**>**Exit Innovator** menu item to close Innovator.

This command logs the user out from all open models, releases all documents and elements, and terminates all model editors.

Sources of Information and Support

This Help, tool tips, documents, hotline and other sources of information are all available to offer you support. MID GmbH also offers extensive training options for getting to know the Innovator products and individual methods used.

Sources of information

- Access the **help** with [F1] to receive context-sensitive support in all active interfaces of the Innovator model editor. From the help topic of the relevant interface, links lead you to the supported procedures.

  Among other things, the help contains information about the following:

  - The installation of Innovator
  - Working with Innovator models in general
  - Special procedures in the models of Innovator products
  - The administration and configuration of Innovator

  You can access help in the Innovator model editor using [F1] or by clicking the question mark to the top right of the ribbon, in all tools in the **Help** menu and via http://help.innovator.de.

- Menu items are described in the **tool tips**. You can see the tool tip showing what the submenu or menu command is used for by expanding the ribbon and hovering the mouse pointer over the menu items. Any shortcut keys are also specified. You can find an overview of the menu items and tool tips in the help system (User Interface (Reference)>Overview of Menu Items).

- Unusual events when running program code in the Innovator interface are logged in the Windows event viewer. It is also possible to have all events of the running interface listed in the **Event Viewer** dialog. Logged events may shine some light on the possible reason for errors in the interface. Therefore please check the event viewer in the event of problems with the interface . Select **Model>Help & Support** and click on the **Event Viewer**.

- The **installation manual** describes the architecture and installation of Innovator as an excerpt of relevant chapters of this Help. You will then be able to create the necessary and desired constellation for Innovator installation and license administration.

  The installation manual is available in both English and German in the online customer service area.

- The **migration manual** contains a description of the work steps necessary for migrating repositories and settings to the current Innovator version.

  The document should provide sufficient information about transforming, data transfer and editing models in the current Innovator version. The migration manual describes how to prepare and carry out transformation for the respective previous versions and informs you of the necessary post-processing steps.

  Please read the Help topic "Installing an Update" to find out the differences between version migration (upgrade) and new release (update).

  The migration manual is available in both English and German as a browse sequence in this Help and in the online customer service area as PDF.

- The **API help** documents the relevant parts of the internal Innovator meta model and the read and change functions available there.

  The API help is available in English only at http://help.innovator.de/SDK/.

- 50 -

> **Note**
>
> There is an extensive range of training options offered by MID GmbH to get to know the Innovator products and individual methods (business process modeling (BPM), business process model and notation (BPMN), unified modeling language (UML), entity relationship modeling (ERM) etc.).

Hotline

Enterprise Edition or **Professional Edition** maintenance customers receive personal support via the Innovator phone hotline and via e-mail.

Telephone support:

Monday to Friday from 9 a.m. until 5 p.m.

Telephone: +49 911 96836-222

You can fax or e-mail your Innovator support:

Fax: +49 911 96836-100

E-mail: support@mid.de