Manatee Installation Guide

March 20, 2025

Contents

This document contains instructions on how to install the Manatee client.

System requirements

The minimum system requirements for a minimal installation on Windows. For more then a minimal installation referer to the sizing guidelines of the Sirenia software components.

- PC OS Versions: Windows 7, Windows 8, Windows 8.1, Windows 10, Windows 11
- Server OS Versions: Windows Server 2016, Windows Server 2019, Windows Server 2022
- .NET 4.5.2 or later (Manatee v1.29) or .NET 4.7.2 or later (Manatee v2)
- vcredist++ 2013 x86 for optical field support (v1.29)
- RAM: 8GB
- Disk space: 4GB minimum recommendation for Windows
- Access to CMR and optionally the CM
- Writable access to `%appdata%`/`Sirenia`/`Manatee`

Resource usages

- Size of installer is ~200Mb, when installed in `%programfiles%` ~500Mb
- `%appdata%` usage is dependent on logging setup, plugins installed and other configuration parameters but would typically be <100Mb (in most cases a lot less)
- Estimated RAM usage is 150-200Mb

Network requirements

The estimated network usage will be highest for CM traffic as CMR happens less frequently. A scenario with 20.000 concurrent users, 10 apps per session and approx 10 transactions pr. user/min. will equal roughly 10Mbit/sec bandwidth requirement on the server side.

Installers

The Driver Platform (Manatee) is distributed as a machine-wide installer (`.msi` file) or as a per-user installer (`.exe` file) as well as a hybrid of the two.

Machine-wide

The machine-wide installer should be used for rolling out Manatee in enterprise environments or in deployments where a normal desktop user does not have administrative rights. Using the machine-wide installer means that:

- The application will *not* self-update.
- All users for a given machine will get the application installed.
- The application will *not* automatically be started when the installer is done.

The installation process is straightforward, simply run the `.msi` file. It should not show any UI and will install shortcuts in the startmenu.

Custom arguments for the msi installer

You can use the following custom arguments when running the `.msi` installer:

Name	Description	Versions
COPYINI	Path to an ini file to copy to the installation directory. The ini file contains default settings.	
COPYSCFG	Path to an encrypted ini file to copy to the installation directory. The ini file contains default settings. You can encrypt the file using the `Crypto.encrypt()` method in a flow.	v2.0+
REGISTRY_USER	A username to use for the CMR/registry. Kwanza)	v2.1+

Name	Description	Versions
REGISTRY_F	A password to use for the CMR/registry. (Kwanza)	v2.1+
REGISTRY_TOKEN	A token to use for the CMR/registry (Kwanza). This is an alternative to using `REGISTRY_USER` and `REGISTRY_PASS`.	
BRANCHSHOF	Specify `yes` to create a shortcut in program files.	
INSTALLDESK	Specify `yes` to create a shortcut on the desktop.	
REGISTERN/	Specify `no` to disable the native messaging host registration. Will prevent the browser extension from working correctly.	

Bundled webview2 runtime

Starting with Manatee v1.29.101, the machine-wide `.msi` installer comes in two flavors: One with a bundled webview2 runtime and one without.

Webview2 is a microsoft component allowing Manatee to use embedded Edge browser views. Microsoft provides separate webview2 installers. For users that already have webview2 installed, the lighter Manatee installer can be safely used.

Per-user

The per-user installer should be used when individual users themselves are responsible for installing software on their own machine. It has the properties that:

- The application will automatically update when new versions are released.
- The application willl only get installed for the one user running the installer.
- The installer will start the application when it has been installed.

The installation procedure is the same as for the machine-wide installer. Once the installer is done Manatee will be started.

Hybrid

The hybrid installer is an `.msi` file which is intended to be run either by each individual user or by an administrator. It will install the application for the each user *when the user next logs in on the machine*

using the per-user installer. Its intended use is for situations where an `.msi` installation is the only viable option (enterprise environments) but the automatic updates are considered critical. e

Staging environments

If you need to stage Manatee installations e.g. some users needs access to a `test` environment others to `prod` then there are a couple of approaches.

Primary group

Each Manatee instance is configured with a *primary group* which determines the flows that are accessible for that given instance. By having e.g. a `PROD` and a `TEST` group and assigning these groups to different flows then you can have some Manatees (with primary group `PROD`) which only have access to *prod* flows and vice versa for `TEST` flows/Manatee instances.

This approach means that you'll be running the same version of Manatee but you'll have made different flows available to different users/machines.

The setting for the *primary group* has the key `ProductionGroup` which is slightly confusing, we admit.

Registry

Another approach is to run two registries and then point each Manatee instance to one or the other - e.g. one registry for *prod* and another for *test*. The *prod* Manatees then need to be configured to access the *prod* registry and likewise for *test*.

In this way you'll have a tighter seperation of flows but you'll still have the same Manatee versions running on all machines.

The setting for the *registry url* is keyed with `Registry`.

Installing other versions

The last approach to staging is to use different installers for different environments. The installers can contain Manatee instances pre-configured with both *primary group* and *registry* but they may also contain different versions of Manatee.

This approach is useful for testing new versions of Manatees and also for keeping configuration changes to a minimum.