

Installation and updates

The Print Service has no installer. The program is delivered as a git repository that contains only the released build. Updates are a `git` checkout, driven by a script that stops the service gracefully first. Configuration, token, job journal and logs live in a separate data folder and survive every update.

Requirements

Operating system	Windows 10/11 or Windows Server 2019 or newer, x64
Runtime	.NET 10 Desktop Runtime (x64) . The build is framework-dependent, the runtime is not bundled.
Git	Git for Windows and read access to the release repository
Network	HTTPS to the VeloxFactory API and WSS to the Reverb server of your instance. With receive mode <code>Polling</code> the API alone is enough.
API token	Token of a user with <code>global:use-api</code> , <code>report-print-task:read</code> and <code>report-print-task:update</code> . <code>printer:read</code> is optional and enables the printer comparison. <code>:read</code> is also the right the private channel checks, a token without it never receives a WebSocket event.
Service account	A dedicated Windows account for the scheduled task, see below
Printers	Installed machine-wide, see Printers and printing

Folder layout

```
C:\kiwisoftware\VeloxPrint\
  app\                clone of the release repository, program only, updated by
                        kiwisoftware.VeloxFactory.PrintServi
  bin\
```

update.ps1

README.md

data\

created by t

config.json

secrets.json

state.db

job journal, outbox, statistics, log buffer, engine status

logs\

Where the data folder is:

1. `--config <path>` sets it explicitly: all files of the instance live next to that configuration file.
2. If the program runs from a folder named `app` or below it, the data folder is `data` next to that `app` folder.
3. Otherwise it is `%ProgramData%\kiwisoftware\VeloxPrint\`.

File	Permissions
<code>config.json</code>	Administrators and service account: modify. Users: read.
<code>secrets.json</code>	Administrators and service account: modify. Users: no access.
<code>state.db</code> , <code>logs\</code>	Service account: modify. Users: read.

The first start with administrator rights creates the data folder and sets these permissions. Setting up the scheduled task adds the service account.

⚠ **Do not copy `secrets.json` to another machine.** The token is encrypted with DPAPI for this machine (scope LocalMachine), so the window of an administrator and the headless service account can both read it. On another machine it cannot be decrypted, enter the token there again.

First installation

1. Create the service account

The scheduled task runs under its own account. It must exist before the task is set up, the Print Service does not create it. PowerShell as administrator:

```
$pw = Read-Host -AsSecureString "Password"
New-LocalUser -Name "svc-veloxprint" -Password $pw -FullName "VeloxFactory Print Service" `
    -Description "Service account VeloxFactory Print Service" -PasswordNeverExpires
UserMayNotChangePassword
```

1. Do not add the account to the Administrators group, the standard Users group is enough.
2. Use `-PasswordNeverExpires`. An expired password stops the task until it is set up again.
3. For printers on a print server in a domain, use a domain account (`DOMAIN\svc-veloxprint`) instead. Only a domain account can authenticate at the print server.
4. Do not use `SYSTEM`. It accesses print servers with the computer account, shares often refuse that.

2. Clone the release

PowerShell as administrator:

```
New-Item -ItemType Directory C:\kiwisoftware\VeloxPrint
git clone https://git.fischernetz.xyz/bfischer/veloxfactory-printservice-release.git
C:\kiwisoftware\VeloxPrint\app
```

3. Configure

1. Start `C:\kiwisoftware\VeloxPrint\app\bin\kiwisoftware.VeloxFactory.PrintService.App.exe` **as administrator**. The first elevated start creates `data\` and restricts its permissions.
2. **Settings → Connection:** base URL of VeloxFactory and API token. **Test connection** checks reachability, token, channel authorization, WebSocket handshake and rate limit headers.
3. **Settings → WebSocket (Reverb):** host, port, scheme and app key of your Reverb server.
4. **Settings → Receiving:** the broadcast ids of this site.
5. Save. See the [Configuration reference](#) for every setting.

4. Set up the scheduled task

Settings → Operation → Set up scheduled task, enter the service account (`.\svc-veloxprint` for a local account) and its password. Alternatively on the command line (as administrator):

```
& "C:\kiwisoftware\VeloxPrint\app\bin\kiwisoftware.VeloxFactory.PrintService.App.exe" --
install-task --user .\svc-veloxprint
```

The setup:

1. grants the account the right *Log on as a batch job*,
2. sets the permissions of the data folder and `secrets.json` for the account,
3. registers the Windows event log source *VeloxFactory Print Service*,
4. creates or updates the task with the settings below.

Task setting	Value	Reason
Name	<code>VeloxFactory Print Service</code> , for an alternative configuration <code>VeloxFactory Print Service (<config name>)</code>	Root folder of the Task Scheduler library
Run	Whether the user is logged on or not, password stored	No desktop needed; without stored password network printers are unreachable
Trigger	At system start, delayed 1 minute	Network and spooler are ready
Action	<code>--headless</code> (plus <code>--config</code> for an alternative instance), start in the program folder	The Task Scheduler would otherwise start in <code>System32</code>
Time limit	None	The Windows default of 3 days would end the engine
On failure	Restart every minute, up to 999 times	Crash recovery
Already running	Do not start a new instance	In addition to the instance mutex
Privileges	Not elevated	Not needed
Battery	Runs on battery, starts when available	Missed starts are caught up

⚠ **Do not edit the task by hand.** Setting it up again from the window or with `--install-task` overwrites it with the values above. After changing the password of the service account, set the task up again, the password is stored in the task.

The window shows the task under **Settings → Operation**: state, account, last and next run, last result and the XML definition. It can start and remove the task there, too.

5. Verify

1. **Printers:** print a test page on every printer the site uses.
2. Start the task (**Settings → Operation → Start** or reboot). The window switches to monitor mode.
3. Create a print task in VeloxFactory for one of the broadcast ids and watch it on the **Dashboard**.

Updates and rollback

PowerShell as administrator:

```
#                                newest                                release
powershell -ExecutionPolicy Bypass -File C:\kiwisoftware\VeloxPrint\app\update.ps1

# a specific release, also the way back to an older one
powershell -ExecutionPolicy Bypass -File C:\kiwisoftware\VeloxPrint\app\update.ps1 -Version 2.0.1
```

Step	What <code>update.ps1</code> does
1. Check	Refuses to run without administrator rights or if <code>app\</code> contains local changes
2. Fetch	<code>git fetch</code> before anything is stopped. If the server is unreachable, the service keeps running. Already on the target: nothing to do.
3. Stop	Calls <code>--stop</code> for every scheduled task that runs from this folder (with its <code>--config</code>). Running print jobs finish, the outbox is flushed. Waits up to 90 s until no process runs from <code>app\bin</code> .
4. Switch	Checks out the newest release (<code>origin/main</code>) or the tag <code>v<Version></code>
5. Start	Starts the scheduled tasks again that were running before

- 1. Releases are git tags (`v2.0.1`). `git -C C:\kiwisoftware\VeloxPrint\app log --oneline` shows the source commit of every release.
- 2. New settings are added to `config.json` with their defaults on the next start.
- 3. Never change files inside `app\`. The script refuses to update a modified folder. Settings belong into `data\`.

Several instances on one machine

Each instance needs its own configuration file. Its name becomes the prefix of all files of the instance, so several instances can share one data folder:

```
data\config.json
data\lager-nord.json
data\lager-nord.secrets.json
data\lager-nord.state.db
data\logs\lager-nord\
```

```
$exe = "C:\kiwisoftware\VeloxPrint\app\bin\kiwisoftware.VeloxFactory.PrintService.App.exe"
& $exe --config C:\kiwisoftware\VeloxPrint\data\lager-nord.json
& $exe --config C:\kiwisoftware\VeloxPrint\data\lager-nord.json --install-task --user .\svc-veloxprint
```

The task of that instance is called `VeloxFactory Print Service (lager-nord)`. Give every instance its own broadcast ids.

Revision #1

Created 2026-09-25 19:46:55 UTC by Benjamin Fischer

Updated 2026-09-25 19:46:55 UTC by Benjamin Fischer