

# Installing Hopp

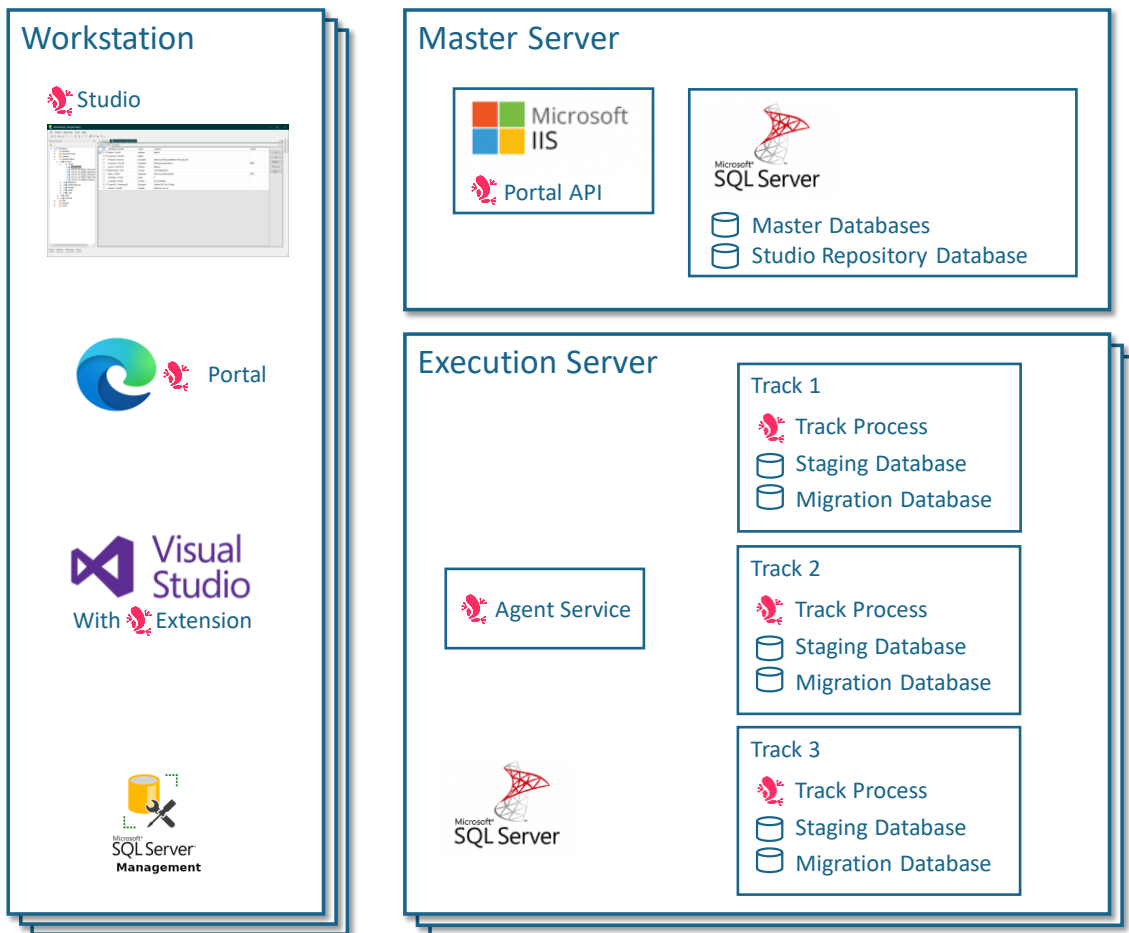
*This guide contains information about how to install and configure an instance of the Hopp migration solution and which preconditions should be met.*

*The guide is available from our download page that also contains links to all the artifacts necessary to install the Hopp components as well as a set of database scripts to setup the Sql Server databases that are the foundation for Hopp.*

## Configuration overview

In its simplest form, Hopp in its entirety can be installed on a single laptop. However, in a real-world scenario with a team collaborating internally to produce mapping in Studio as well as externally with the business users involved in the migration, the setup will often be more elaborate.

The figure below shows the full Hopp setup.





<i>Master Server</i>	A single Master Server typically contains the database serving as repository for the Studio and the Master Database of the Director Runtime, containing housekeeping of all Migration Servers and Tracks. The Master Server also hosts the Tracking web application on an Internet Information Server instance
<i>Execution Server</i>	One or more Execution Servers contain the Runtime libraries and Services as well as the databases and folders necessary to run migration iterations for separate migration projects in one or more Tracks. The Master Server may act as a Migration Server as well in a single server set-up
<i>Workstation</i>	<p>Users directly working on the migration project have a Workstation with these local installations.</p> <ul style="list-style-type: none"><li>• The Studio so they can collaborate on the mapping. Users all connect the Studio to the same Mapping Repository Database residing on a Master Server</li><li>• The Engine Framework containing the code generators so they can get the latest version of the mapping, publish it from the Studio, and use the code generator to generate new engine code</li><li>• Visual Studio to implement manual rules and build and deploy migration engines in the runtime environment</li><li>• A Web browser to access the Hopp Portal in order to execute migration jobs, view the migration events and results, and to collaborate on these</li></ul>
<i>Business PC/Tablet</i>	Users working outside the migration project, typically involved in the testing and quality assurance, only need a web browser access the Hopp Portal to view the migration events and results and to collaborate on these

## Single Server setup

It is common and indeed advisable to start out in a simpler, single server setup. In this setup a single server acts as both Master Server and Execution Server.

Team members and Business users have the migFx client components installed on their own machines and use the migFx client components (mainly Studio and Director) locally.

- The local instance of Studio will communicate with the Repository database on the Master Server via an Sql connection
- The local instance of Director will communicate with the Director Runtime via Web Service on the Master Server

## Single Machine Setup

An even simpler setup is to install all Hopp on one, single machine. Users will then have to connect to this machine using a remote desktop connection and run the Workstation components on the machine itself.

A single machine setup is especially useful in a Proof of Concept installation.



---

## Preconditions

<i>Service Account</i>	<p>The Hopp Runtime needs to run under a service account. In a single server setup, this can be a local account on the server. In a multi-server setup, this must be an Active Directory account.</p> <p>The service account must have the roles necessary to act as Local Administrator on the servers involved.</p> <p>Can be a Managed Service Account (MSA) to avoid password handling</p>
<i>Certificates</i>	<ul style="list-style-type: none"><li>• A trusted TLS certificate for the https binding on the web server</li></ul>
<i>Network</i>	<p>The Master Server must open the necessary ports for</p> <ul style="list-style-type: none"><li>• Sql Server (default is 1433)</li><li>• https (443) depending on Web Server configuration</li><li>• File sharing</li></ul> <p>The Migration Server must open the necessary ports for</p> <ul style="list-style-type: none"><li>• Sql Server (default 1433)</li></ul>
<i>Disk</i>	<p>Both the Master Server and the Migration Server should have a D: drive dedicated to Hopp.</p> <ul style="list-style-type: none"><li>• Master Server: Minimum 25GB</li><li>• Migration Server: Minimum 250GB, larger depending on data volume of migration</li></ul>

## Pre-installation check list

These activities typically takes a bit of time and can be initiated prior to installation

- Create the Hopp Service Account
- Create the trusted TLS certificate for https binding

---

## Third party software

In addition to the software components of Hopp, several other, third-party software must be installed.

<i>Microsoft Sql Server</i>	Install on Master and Execution Server. Express Edition is sufficient for smaller migrations. Larger migrations may benefit from a licensed instance capable of utilizing a higher number of CPUs.
<i>Microsoft Sql Server Management Studio</i>	Install on Workstation of team members that will work on Studio Source Maps and operators in charge of maintaining Hopp
<i>Visual Studio</i>	Install on Workstation of team members that will implement manual rules. Minimum Visual Studio 2022, Community Edition is sufficient



<i>Net SDK</i>	Install the Net SDK on all Master and Execution Servers. Version must correspond to the current version used by the Hopp solution as indicated on the download page.
<i>Net Hosting Bundle</i>	The dotnet hosting bundle must be installed on the Master server. Version must correspond to the current version used by the Hopp solution as indicated on the download page.

---

## Sql Server installation

Sql Server must be installed on the Master Server and the Migration Server

- Install Windows Authentication and/or Sql Server Authentication
- Add the Hopp Service Account
  - either as SysAdmin (simplest)
  - or as BulkAdmin and with db\_owner on all databases

### Separate App and Database servers

It is recommended to install Sql Server on Master- and Execution Servers. In some installations, policies may require application- and database servers to be separated.

If databases are not installed on the Execution Server, these additional requirements must be met:

- In the absence of a full Sql Server installation, the Sql Server command utility tools must be installed on the execution servers (specifically the BCP bulk load utility)
- A file share must be created on the database server for files to be bulk loaded into the Staging Databases. The Hopp Service Account must be granted full control to this file share
- In a setup where multiple execution servers are connecting to one database server, the track-specific databases should be named accordingly (database name should include the name of the execution server)

### Data protection and failover considerations

The Hopp installation is normally somewhat of a hybrid in comparison to the policies applied in most installations. The data being migrated certainly are in many cases sensitive and require production level data protected and access restrictions. At the same time, the Hopp migration software is not part of the normal production setup in terms of availability, stability and failover protection.

If possible, it is recommended to avoid production-like setups like mirroring and clustering in order to maximize the migration throughput. In case of a complete and catastrophic failure of the database environment, the file based nature of the source data inputs to the Hopp migration software makes it a normally trivial task to re-establish the migration environment and simply run the migration again.

---

## Setting up the Master Server

### Configure Folders

Recommended folder structure on Master Server



- **D:\MigFx**
  - **Database:** Master, Repository and Portal database files here
    - **Project:** Project Database files here (when setting up a new migration project)
  - **Runtime: OBS:** The MigFx Service Account must have full control access to this folder
    - **Engine:** Hopp will deploy migration engines here
    - **Director:** The Director Runtime will be installed here
    - **Temp:** Director will use this folder for temporary files
  - **Web:** Root folder for the IIS default web site

## Create Databases

Install Sql Server as per previous guidelines

**NB:** Please ensure that all databases are created with the same Sql Server collation.

- Create database MigFx\_Master and run setup script MigFx.MasterDb.Setup.sql
- Create database MigFx\_Repository and run setup script MigFx.RepositoryDb.Setup.sql
- Create database MigFx\_Portal and run setup script MigFx.PortalDb.Setup.sql

## Install Hopp components

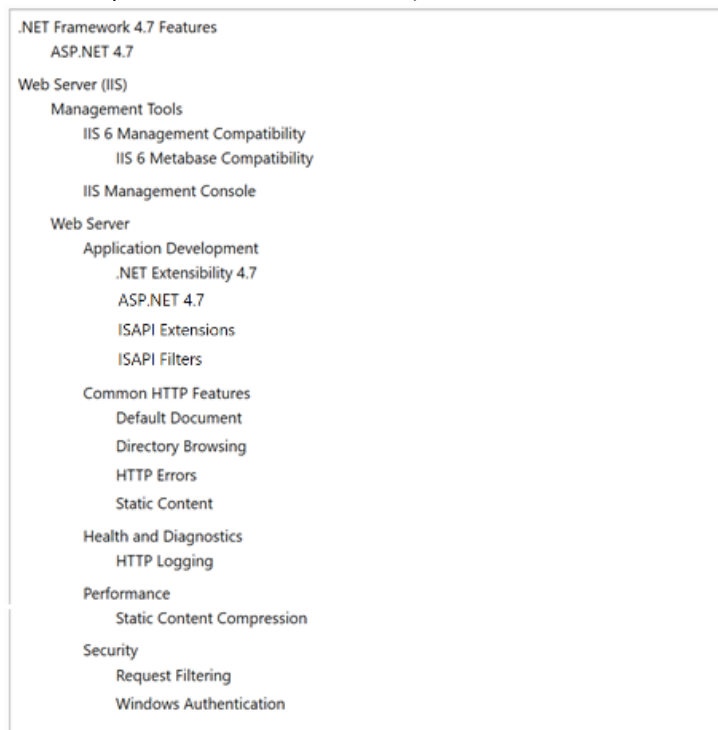
- Run the MigFx.Director.Runtime.msi installer

If your folder structure follows the recommendation, you will not have to change any install locations.



## Configure IIS

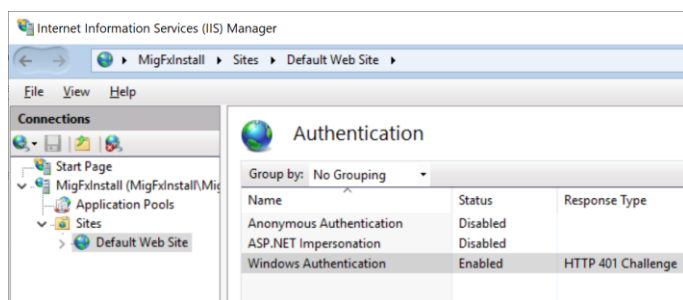
- Use Server Manager to install IIS with this configuration (.NET version numbers in screenshot may be obsolete, please use newest available).



- Install the .NET Core Hosting Bundle - <https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/iis/hosting-bundle>

## Authentication

- Anonymous Authentication must be enabled
- Windows Authentication must be Enabled



## Physical Path

It is recommended to set the physical path of the Default Web Site to the *Web* folder in the MigFx folder hierarchy (see above).

## Application Pool

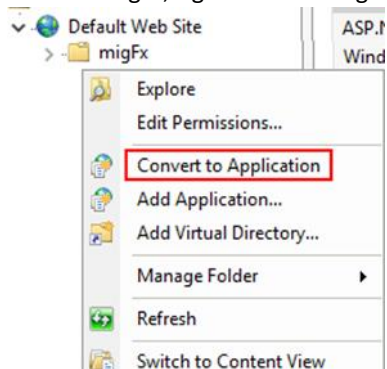
It is preferable to run the Portal Web Application under a separate, dedicated application pool. Suggested name: *Hopp.Portal*

Configure the App Pool to run under the Hopp Service Account.



## Configure the Portal

- From the Portal.zip file, copy the *migFx* folder to the physical path of the Default Web Site
- In IIS Manager, right click the migFx folder under the Default Web Site and *Convert to Application*



- Edit the appsettings.json file
  - Be careful when editing json, especially remember that any backslash must be duplicated for the json to parse correctly
  - Set connection strings
    - *MigFxPortal*: The MigFx\_Portal database created above
    - *MigFxMigration*: The MigFx\_Master database created above
  - In the *Authentication* section:
    - Set the *PortalUrl* to the url that will be used to access the portal
    - Set *IdentityProvider* to *Windows* or *MigFx* as appropriate
  - For first time setup, uncomment and edit the seed user to create the first user in the portal. Once created, the seed user should be commented out or removed from appsettings.json
- Edit wwwroot/Index.html
  - Set the base href to migFx: `<base href="/migfx/" />`

### *Optional: Create and use a certificate for signing of JSON Web Tokens*

By default, the Portal will create and use a transient certificate to sign the JSON Web Tokens of authenticated users.

This is normally sufficient.

However; In some scenarios, it may be required to use a persisted certificate for the signing of JSON Web Tokens.

In this case, you can create and install a certificate and reference this from appsettings.json:

- The certificate must be created with an RSA private key ([Creating RSA Keys using OpenSSL](#))
  - These are the openssl command to create a certificate with a 1-year expiry

```
openssl genrsa -out private-key.pem 3072
openssl rsa -in private-key.pem -pubout -out public-key.pem
openssl req -new -x509 -key private-key.pem -out cert.pem -days 360
openssl pkcs12 -export -inkey private-key.pem -in cert.pem -out cert.pfx
```



- Install the certificate on the master server. Recommended install location is *Local Computer/Personal*:



- In the *Authentication* section of the appsettings.json:
  - Uncomment and update the *SigningCertificate* section to identify the certificate installed above

---

## Setting up the Execution Server

### Configure Folders

Recommended folder structure on Execution Server

- **D:\MigFx**
  - **Database:** Database files for databases global to the migration server
  - **Runtime:** The MigFx Service Account must have full control access to this folder
    - **Track** (one sub folder pr track on the Migration Server)
      - **01**
        - **Database:** Place the MigFx\_01\_Migration and MigFx\_01\_Staging database files here
        - (The MigFx Runtime will automatically create other, required folders for the Track here)
      - **02 ....**
      - **XX**
        - **Database:** MigFx\_XX\_Migration and MigFx\_XX\_Staging database files

### Create Databases

Install Sql Server as per previous guidelines.

**NB:** Please ensure that all databases are created with the same Sql Server collation.

- Create database MigFx\_UTILITY
  - Database files in D:\MigFx\Database
  - Run setup script MigFx.UtilityDb.Setup.sql
- For each Track XX:

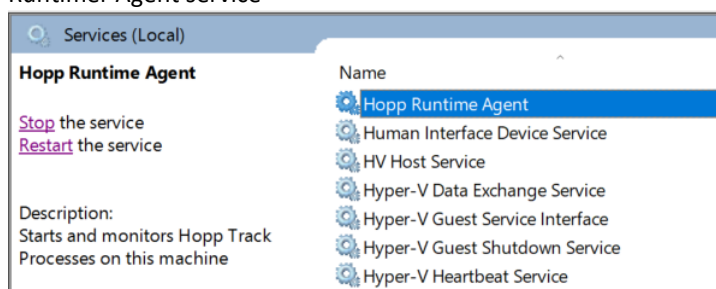




- Note: In a setup where multiple execution servers are connecting to one database server, the track-specific databases below should be named accordingly (database name should include the name of the execution server)
- Create database MigFx\_XX\_Migration
  - Database files in the D:\MigFx\Runtime\Database\XX\Database
  - Run setup script MigFx.MigrationDb.Setup.sql
- Create database MigFx\_XX\_Staging
  - Database files in the D:\MigFx\Runtime\Database\XX\Database
  - (no setup script, database is maintained automatically by MigFx)

## Install Hopp Components

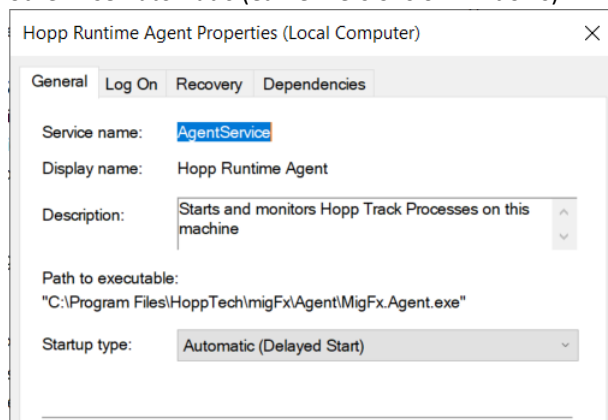
- Run the MigFx.EngineFramework.msi installer
- Run the MigFx.Director.Agent.msi installer
  - Navigate to the installation folder (normally C:\Program Files\HoppTech\migFx\Agent) and edit the MigFx.Agent.dll.config file:
    - *MigFxMaster* connection string must connect to the MigFx\_Master database on the Master Server
    - Change *RuntimePath* to point to the Director folder on the master server (local path is single master install, Network Uri (starting with \\) if multi-server setup
    - If you need to access the RuntimePath with credentials, set *Credentials* to the credentials as needed. If not, you can delete this element
  - Change the service to run under the Hopp Service Account
    - Open the Services Control Panel add-in (Windows-R, services.msc) and locate the Hopp Runtime Agent service



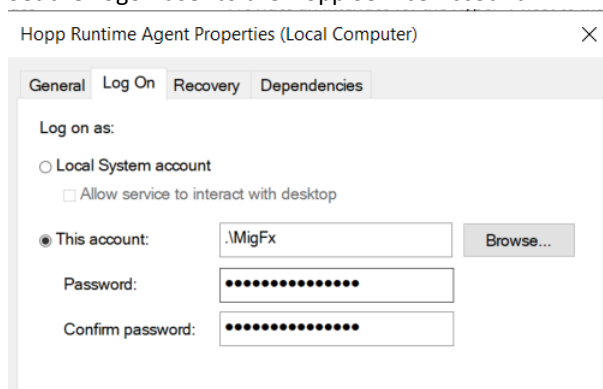
- Open the Properties dialog for the service



- Ensure that Startup mode is set to *Automatic (Delayed Start)* if available, otherwise *Automatic* (earlier versions of Windows)



- Set the Logon user to the Hopp Service Account



---

## Setting up the Workstation

### Visual Studio

- Minimum Visual Studio 2022 any edition, Community Edition is sufficient
- At least include the .NET desktop development workload
  - After install of Visual Studio, install the MigFx Visual Studio extension VsExtension.2022.vsix
- Git or other source control infrastructure

### Sql Server Management Studio

- No special instructions

### Install Hopp Components

- Run the MigFx.Libraries.msi installer
- Run the MigFx.Studio.exe setup bootstrapper
  - The bootstrapper will install any missing prerequisites (SAP Crystal Reports and Sql Server LocalDb) and migFx Studio



- Run the MigFx.EngineFramework.msi installer