

Installing migFx

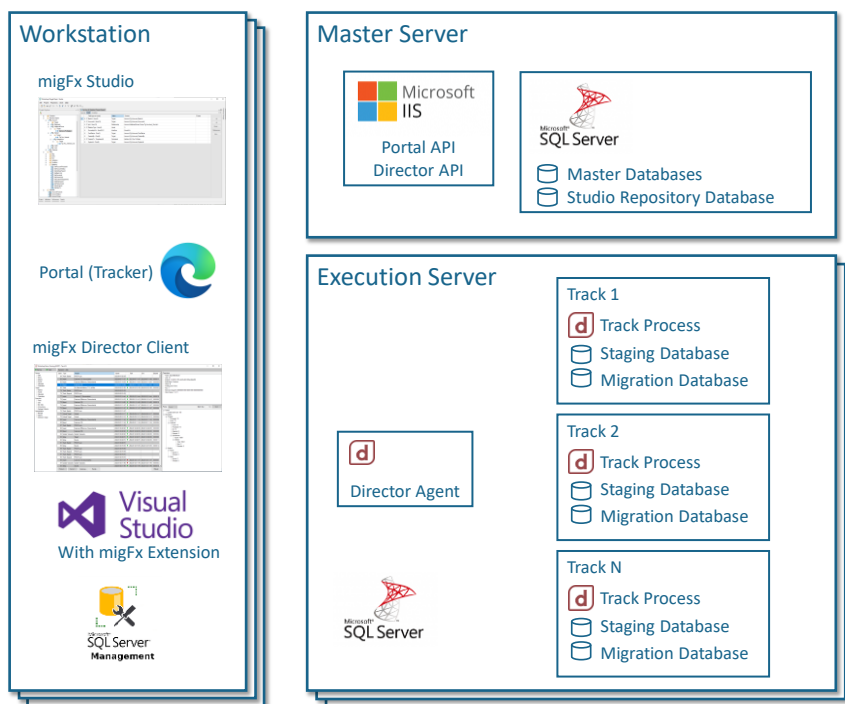
This guide contains information about how to install and configure an instance of migFx and which preconditions should be met.

The guide is accompanied by a set of installers to install the migFx components as well as a set of database scripts to setup the Sql Server databases that are the foundation for migFx.

Configuration overview

In its simplest form, migFx 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 migFx setup.



Master Server 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

Execution Server One or more Execution Servers contain the Director 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



- Workstation* Users directly working on the migration project have a Workstation with these local installations.
- 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
 - 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
 - Visual Studio to implement manual rules and build and deploy migration engines in the runtime environment
 - The Director Client to initiate and monitor jobs in the Director Runtime
 - A Web browser to view the migration events and results and to collaborate on these
- Business PC/Tablet* Users working outside the migration project, typically involved in the testing and quality assurance, only need a web browser 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 migFx on one, single machine. Users will then have to connect to this machine using a remote desktop connection and run the migFx Workstation components on the machine itself.

Preconditions

- Service Account* The migFx 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.
- The service account must have the roles necessary to act as Local Administrator on the servers involved.
- Network* The Master Server must open the necessary ports for
- Sql Server (default is 1433)
 - http (80) and/or https (443) depending on Web Server configuration
 - File sharing
- The Migration Server must open the necessary ports for



- Sql Server (default 1433)

Disk

Both the Master Server and the Migration Server should have a D: drive dedicated to migFx.

- Master Server: Minimum 25GB
- Migration Server: Minimum 250GB, larger depending on data volume of migration

Third party software

In addition to the software components of migFx, 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 migFx
<i>Visual Studio</i>	Install on Workstation of team members that will implement manual rules. Minimum Visual Studio 2017, Community Edition is sufficient

Sql Server installation

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

- Install Mixed Mode authentication (Windows Authentication and Sql Server Authentication)
- Add the MigFx service Windows Account as SysAdmin
- Create a Sql Server login as well with SysAdmin authorization. Recommended MigFx.Sql, password MigFx

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:** migFx 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. You may need to include LocalDb in the installation, see **Error! Reference source not found.**

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

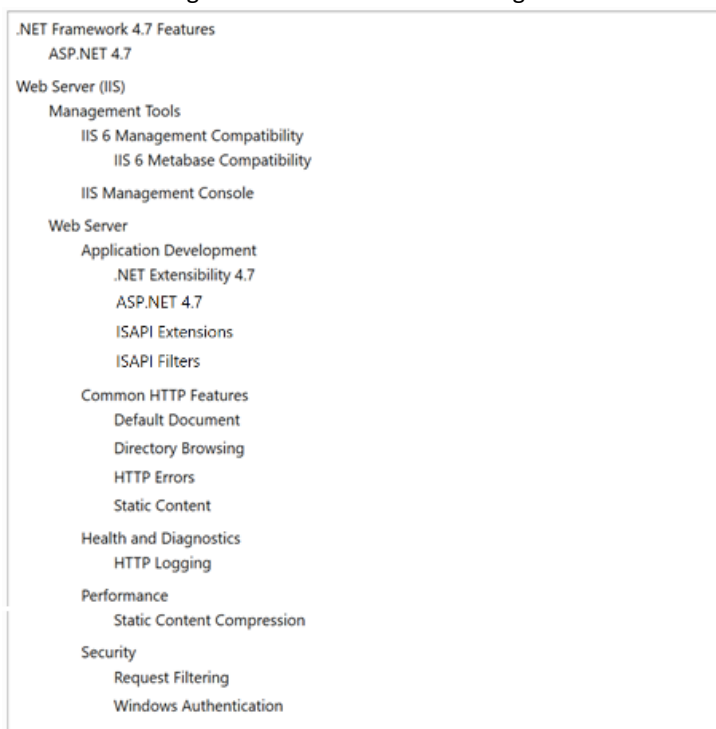
Install migFx components

- 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

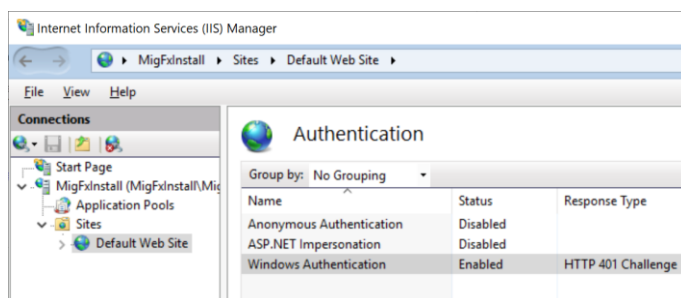
- Use Server Manager to install IIS with this configuration.



- 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).



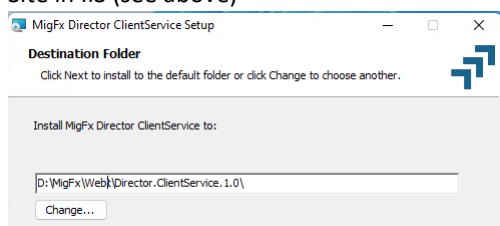
App Pools

It is preferable to run the migFx Director services and the Portal Web Application under separate, dedicated application pools. Suggested names:

- migFx.Director
- migFx.Portal

Configure Director Client Service

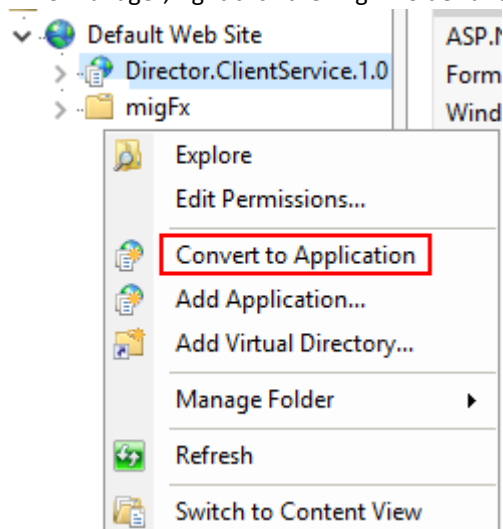
- Run the MigFx.Director.ClientService.msi installer
- The installation path should be to a *Director.ClientService.1.0* under the Physical Path for the Default Web Site in IIS (see above)



- Edit the Director.ClientService.1.0\web.config file
 - Change the MigMaster and MigPortal connectionstrings as appropriate
 - Change the FileTransferURN app setting to the Runtime/Temp Folder (see above)

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
 - Set the *Hosting/BasePath* to the url that will be used to access the portal



- Set Identity provider 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/" />`

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.

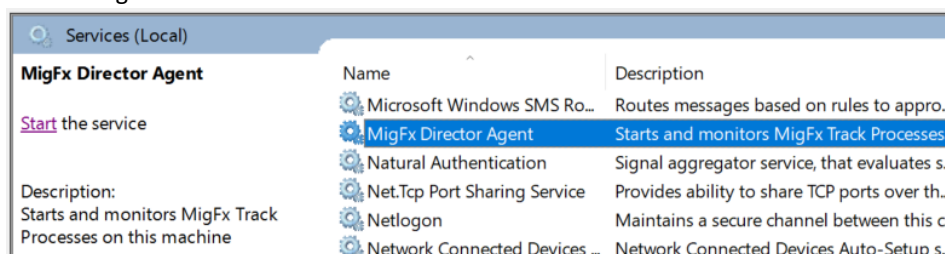
NB: Please ensure that all databases are created with the same Sql Server collation.

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

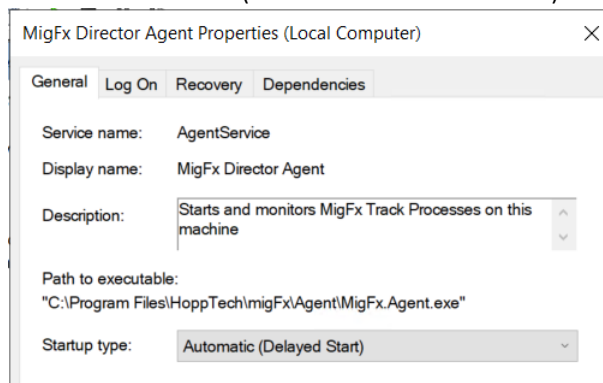


migFx Components

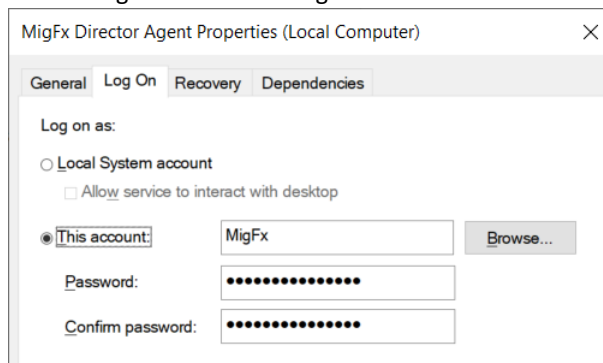
- Run the MigFx.EngineFramework.msi installer
- Run the MigFx.Director.Agent.msi installer
 - Navigate to the installation folder (C:\Program Files\HoppTech\migFx\Agent) and edit the MigFx.Agent.dll.config file:
 - *MigFxMaster* connection string must connect to the MigFx_Master database on the Master 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 MigFx Service Account
 - Open the Services Control Panel add-in (Windows-R, services.msc) and locate the MigFx Director 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 MigFx Service Account





Setting up the Workstation

Visual Studio

- Minimum Visual Studio 2017 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
 - Visual Studio 2017 and 2019: VsExtension.2019.vsix
 - Visual Studio 2022 and higher: VsExtension.2022.vsix
- Git or other source control infrastructure

Sql Server Management Studio

- No special instructions

migFx Components

- Run the MigFx.Libraries.msi installer
- Run the MigFx.Director.Client.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
 - Alternatively, install SqlLocalDb and Crystal Reports separately and then install Studio from the MigFx.Studio.msi installer package.
 - Crystal Reports Runtime can be retrieved here:
https://hopp.tech/doc/resources/CRRuntime_64bit_13_0_23.msi
 - SqlLocalDb 2019 can be retrieved here:
<https://hopp.tech/doc/resources/SqlLocalDb.2019.msi>
- Run the MigFx.EngineFramework.msi installer