

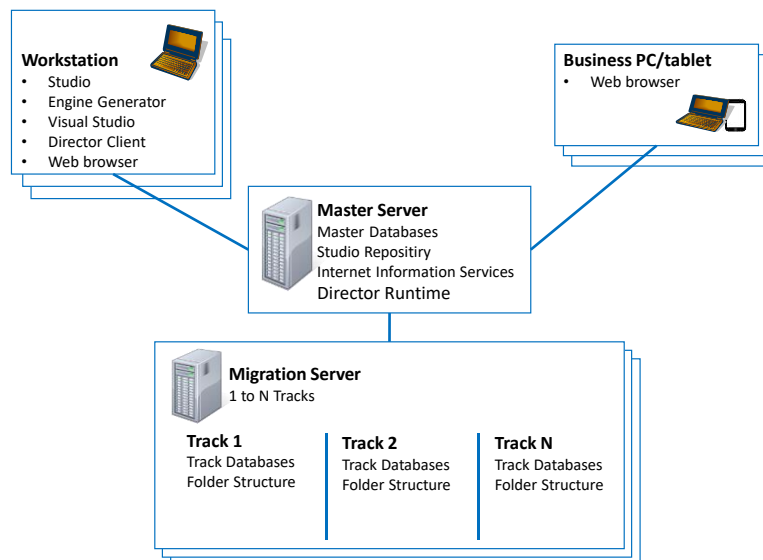
Installing migFx

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.



<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>Migration Server</i>	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
<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"> 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 Migration 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 of 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)
- Visual Studio Remote Debugger connection



Workstation, if used with Visual Studio to implement migFx rules

- Inbound connection from Visual Studio Remote debugger

Disk

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

- Master Server: Minimum 25GB
- Migration Server: Minimum 50GB, 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>Internet Information Server</i>	Install (activate) on Master Server
<i>Visual Studio</i>	Install on Workstation of team members that will implement manual rules. Minimum Visual Studio 2017, Community Edition is sufficient
<i>Visual Studio Remote Debugger</i>	Install on Migration Server for Multi- or Single Server setup. Not required for a Single Machine setup

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 SchemaBackup database files here
 - **Project:** Project Database files here (when setting up a new migration project)
 - **Runtime:** 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

Team members will deploy migFx engines from their Workstations to subfolder in D:\MigFx\Runtime\Engine. Consider creating a share for this folder with public access.

Create databases

Install Sql Server as per previous guidelines. You may need to include LocalDb in the installation, see below.

- Create database MigFx_Master and run setup script
MigFx.Installation\Database\MigFx.MasterDb.Setup.sql
- Create database MigFx_Repository and run setup script
MigFx.Installation\Database\MigFx.RepositoryDb.Setup.sql
- Create database MigFx_SchemaBackup and run setup script
MigFx.Installation\Database\MigFx.SchemaBackupDb.Setup.sql

To create the migFx authorization database, run the program MigFx.AzManUtility.exe located in the MigFx.Installation\Database\MigFx.AzManUtility folder to create a database file *.mdf). Move this file to the database data directory and attach it using the ATTACH command provided by the utility.

It is important to use this utility, as it allows you to be added as a MigFx Administrator giving you the authorizations necessary to setup MigFx.

Should I install LocalDb?

The AzMan Utility uses LocalDb to create the database mdf file, so LocalDb must be installed for this utility to work.

If you are setting up migFx on a single machine, you can at this point install the migFx Studio, as this install will automatically download and install LocalDb for you.

If you are setting up the Master Server in a Single Server or Multi Server setup, you should include LocalDb in the Sql Server installation.

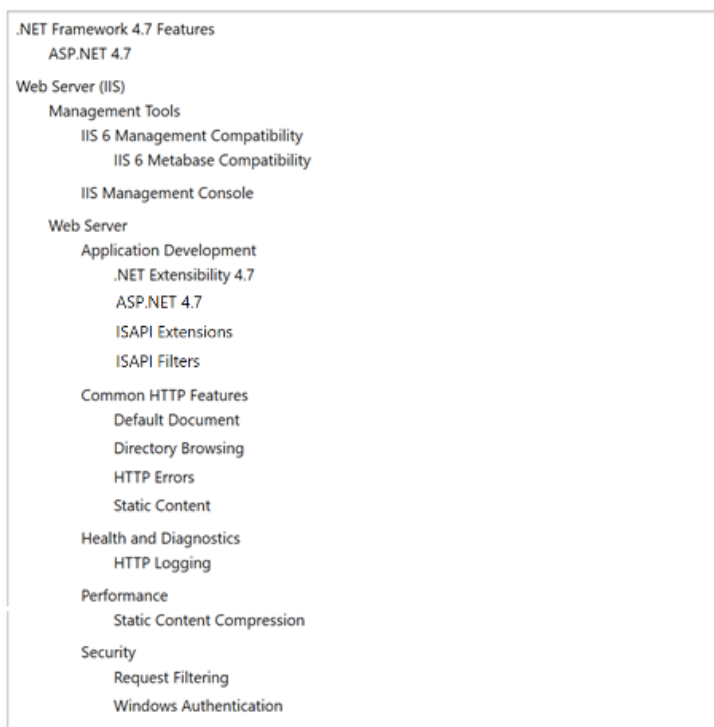
Install migFx components

- From the folder MigFx.Installation\Installers
 - Run the MigFx.Interfaces.x64.msi installer
- From the folder MigFx.Installation\Installers\MasterServer
 - Run the MigFx.Director.Runtime.x64.msi installer

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

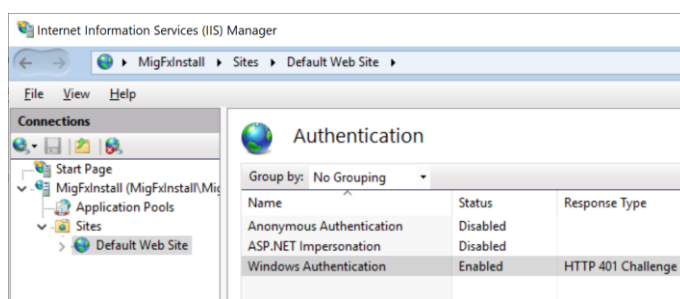
Configure IIS

Use Server Manager to install IIS with this configuration.



Authentication

- Anonymous Authentication must be disabled
- Windows Authentication must be Enabled



Physical Path

Change the Physical Path of the default web site to D:\MigFx\Web

Web.config

Copy the file MigFx.Installation\IIS\master.web.config to the physical path of the default web site (above) and rename it to web.config.

Edit as appropriate:

- Change connection strings (MigFx_Master and MigFx_AzMan) if necessary
- Change the Application User and Domain Query User to the MigFx Service Account

App Pools

It is preferable to run the migFx Director services and the Tracker Web Application under separate, dedicated application pools.



These application pools are supplied in the can be imported into IIS from a command prompt (cd to the MigFx.Installation\IIS folder) with this command:

```
%windir%\system32\inetsrv\appcmd add apppool /in < MigFx.AppPools.xml
```

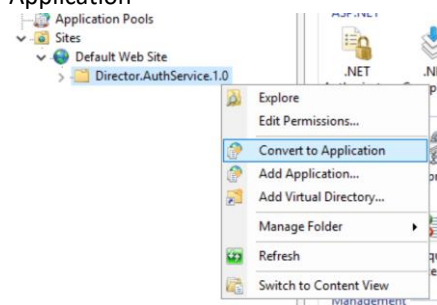
This will import the *MigFx.Director* and *MigFx.Tracker* app pools with the correct configuration for migFx

Configure Web Applications

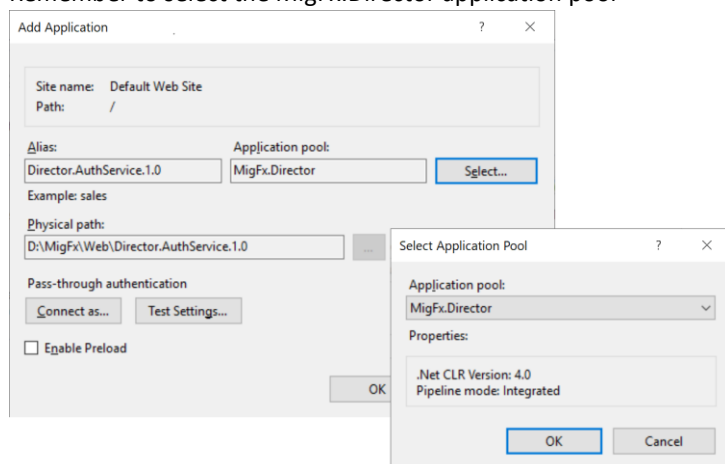
Director AuthService

- Copy the folder MigFx.Installation\IIS\Director.AuthService.1.0 to the root folder of your IIS default web site (D:\MigFx\Web)
- In Internet Information Services manager, the folder should show up. Right click and Convert to

Application



- Remember to select the MigFx.Director application pool



- It should not be necessary to edit the web.config file

Director ClientService

- Copy the folder MigFx.Installation\IIS\Director.ClientService.1.0 to the root folder of your IIS default web site (D:\MigFx\Web)
- As before, convert the folder to an Application and remember to select the *MigFx.Director* app pool. You may have to refresh the list of applications in the IIS manager
- Edit the web.config file
 - Change the httpRuntime/authorization/allow tag to point to the MigFx Service Account (do not use dot (.) for domain name)



Tracker

- Copy the folder MigFx.Installation\IIS\Tracker to the root folder of your IIS default web site (D:\MigFx\Web)
- As before, convert the folder to an Application and remember to select the *MigFx.Tracker* app pool (OBS: not the MigFx.Director app pool). You may have to refresh the list of applications in the IIS manager
- Edit the web.config file
 - Change the MigSchemaBackup connection string as appropriate

Setting up the Migration Server

Configure Folders

Recommended folder structure on Migration 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)
 -
 - **XX**
 - **Database:** MigFx_XX_Migration and MigFx_XX_Staging database files

Team members will deploy migFx engines from their Workstations to subfolder in D:\MigFx\Runtime\Engine. Consider creating a share for this folder with public access.

Create Databases

Install Sql Server as per previous guidelines. It is not necessary to include LocalDb in installation

- Create database MigFx_Utility
 - Database files in D:\MigFx\Database
 - Run setup script MigFx.Installation\Database\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.Installation\Database\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

- From the folder MigFx.Installation\Installers
 - Run the MigFx.Interfaces.x64.msi installer
- From the folder MigFx.Installation\Installers\MigrationServer
 - Run the MigFx.Director.Server.PerfCounters.x64.msi installer
 - Run the MigFx.EngineFramework.Server.x64.msi installer
 - **OBS:** For Single Machine setup: Run the MigFx.EngineFramework.Workstation.x86.msi installer from the folder MigFx.Installation\Installers\Workstation
 - Run the MigFx.Director.Server.Agent.x64.msi installer
 - Use the MigFx Service Account as credentials when installing the Agent
 - Navigate to the installation folder (C:\Program Files\HoppTech\MigFx.Director.TrackAgent\1.0) and edit the MigFx.Agent.exe.config file:
 - *MigFxMaster* connection string must connect to the MigFx_Master database on the Master Server
 - Change *DirectorShareCredentials* to the credentials of the MigFx Service Account
 - Change *DirectorSharePath* to point to the Director folder on the master server (local path is single master install, Network Uri (starting with \\) if multi-server setup)
- Go to Windows Services and start the service MigFx.Director.Agent.1.0
 - In properties, set startup mode to Automatic

Setting up the Workstation

Visual Studio

- At least include the .NET desktop development workload
- After install, from the folder MigFx.Installation\Installers\WorkStation
 - Run the Visual Studio Extension installer VsExtension.vsix
- Git or other

Sql Server Management Studio

- No special instructions

migFx Components

- From the folder MigFx.Installation\Installers



- Run the MigFx.Interfaces.x64.msi installer
- From the folder MigFx.Installation\Installers\WorkStation
 - Run the MigFx.Director.x64.msi installer
 - Run the MigFx.Studio.x64.exe setup
 - The setup will download and install prerequisites, if not already present (SAP Crystal Reports and Sql Server LocalDb)
 - Run the MigFx.EngineFramework.Workstation.x86.msi installer