**EDAM Studio** Installation Guide

2023-01-21

# How to Install EDAM Studio

The current version of EDAM requires to do some previous configuration and installation of resources by hand before running the application for the first time.

## Step 1.0 Preparing the EDAM Database

Make sure to have MS-SQL Server installed before going through the installation steps. It has been tested in 2019, and 2022 Developer, Express, Local and Community versions. Once SQL Server is available open the EDAM Database project, run publishing it to the installed server. Make sure to update the related connection string as required in the next Step.

### Step 1.1 Add Reference Data

Publishing the database does not run the scripts to insert code-sets, go to the “Scripts” folder and run all scripts related to reference data.

## Step 2.0 Editing Settings File appsettings.json

Visit the Edam.Consoles/Edam.WinUI folder to find the appsettings.json file. In this file there are just a few key values to edit. A short discussion of keys and values follow:

|  |  |  |
| --- | --- | --- |
| **Key** | **Default Value** | **Description** |
| DefaultApiScope | Local | Value can be Local or Remote stating if there is an API or middle tier (Remote). |
| DefaultUrlSource | <https://localhost:7179/> | If there is a middle tier, this will be its URL |
| ReferenceDataTemplatesFolder | ReferenceDataTemplates | For future use. |
| ReferenceData/  ConnectionStringKey | RefDataLocal | Points to the connection string key id to use for Reference Data (see ConnectionStrings section ahead). |
| HomeControl | ProjectView | Upon application start-up this is the landing menu item. Other landing sites may be offered later. |
| AM.DB.Key | Kifdbv6r0 | Deprecated |
| DefaultInPath | ApplicationData/AM\_Console/Samples/ | Deprecated |
| DefaultOutPath | ApplicationData/Temp | Deprecated |
| DefaultTextMapFolder | ../../TextMaps/ | See 2.1 section ahead. |
| AssetConsolePath | /ApplicationData/Edam.App.Data/ | Default Application Data folder containing initial resources. |
| AssetProjectsPath | /Projects/ | Relative path for the Projects folder, for now don’t change. |
| AssetArgumentsTemplatePath | Templates/ToAssets.Args.json | Path to the project arguments JSON file. |
| VaultSecretsScope | LOCAL | See 2.2 section ahead. |
| VaultAssemblyAndTypeKey | See sample in section 2.2 | See 2.2 section ahead. |
| EditorLanguageMapFileName | EditorLanguageTextMap.json | Monaco Code Editor language(s) JSON file. |
| DefaultCodeEditorKey | CodeEditorUrl | Key that defines the Code Editor URL. |
| CodeEditorUrl | See sample in Step 2.3 | See 2.3 section ahead. |
| IdentityScope | Local | Identity will be managed locally (keep this value). |
| IdentityConnectionKey | N/A | Since scope is “Local” no need to specify. |
| DefaultDatabaseKey | N/A | Default database key as defined in the connections string section. |
| AssetDataPersistFolder | Temp | Relative path location to store application temporary working files and other stuff. |
| EdamSettingsFileName | Edam.Settings.json | See Step 3.0 section. |

The previous key – values will help to quickly test the application locally.

### Step 2.1 Mapping Keywords, Types and other

The “DefaultTextMapFolder” key states the relative path to the text maps folder. These are used to map source to target types, keywords, and values. Supported types, keywords or other for a given schema language including JSON, GraphQL, SQL, XSD, DDL, B2B (EDI) or other.

In your “.Args.” projects file the specific map to use should be defined.

### Step 2.2 Locating Security Key Vault Assembly

Encryption keys are stored somewhere and the “Edam.Security” library project defines classes and methods to support retrieving those. By default, a crude implementation is provided that could or should be redefine when possible. Once the library (“dll”) is provided its signature can be changed on the configuration file replacing:

Edam.Security, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null;Edam.Security.SecurityVault.Keys.SecurityKeys

The “VaultSecretSope” Identity can be managed locally or remote. Keep the default value and in future documentation updates options will be provided.

### Step 2.3 Locating the Microsoft Monaco Code Editor

EDAM uses Microsoft Monaco Code Editor and therefore the application needs to know where the code is. The “CodeEditorUrl” should contain the partial path location as follows:

\\web\\monaco-editor/code-editor.html

Nothing needs to be changed since the path is out of the box the partial path needed to find the Editor resources.

## Step 3.0. Editing Edam.App.Data Edam.Settings.json

First remove the following are within the file.

      {

         "Name": "AM.Console",

         "Type": "ConsolePath",

         "UriText": "c:/prjs/AppData/Edam.AM.Console/"

       }

The EDAM application templates, and projects data should be found somewhere, to provide a quick start a sample “App.Data” with some Health Care (HC) projects are provided. Upon inspection this folder contains a collection of folders, and the Projects is one of them. As of current release the Projects folder contains the HC Communicable Diseases and may contain other samples. Review its content since every other project should or may be similar.

Alternatively, it is possible to add multiple “App.Data” folders as shown in the segment that is being removed as specify above. To quickly get you started just copy the content of the proved sample and copy it wherever, then add a section to specify its location. The application will display the alternate locations in a drop-down box that will include those define in this file.

## Application Package

Don’t look for an execution file but build a deployment package that could be used to publish the application using “Click Once” or other method.

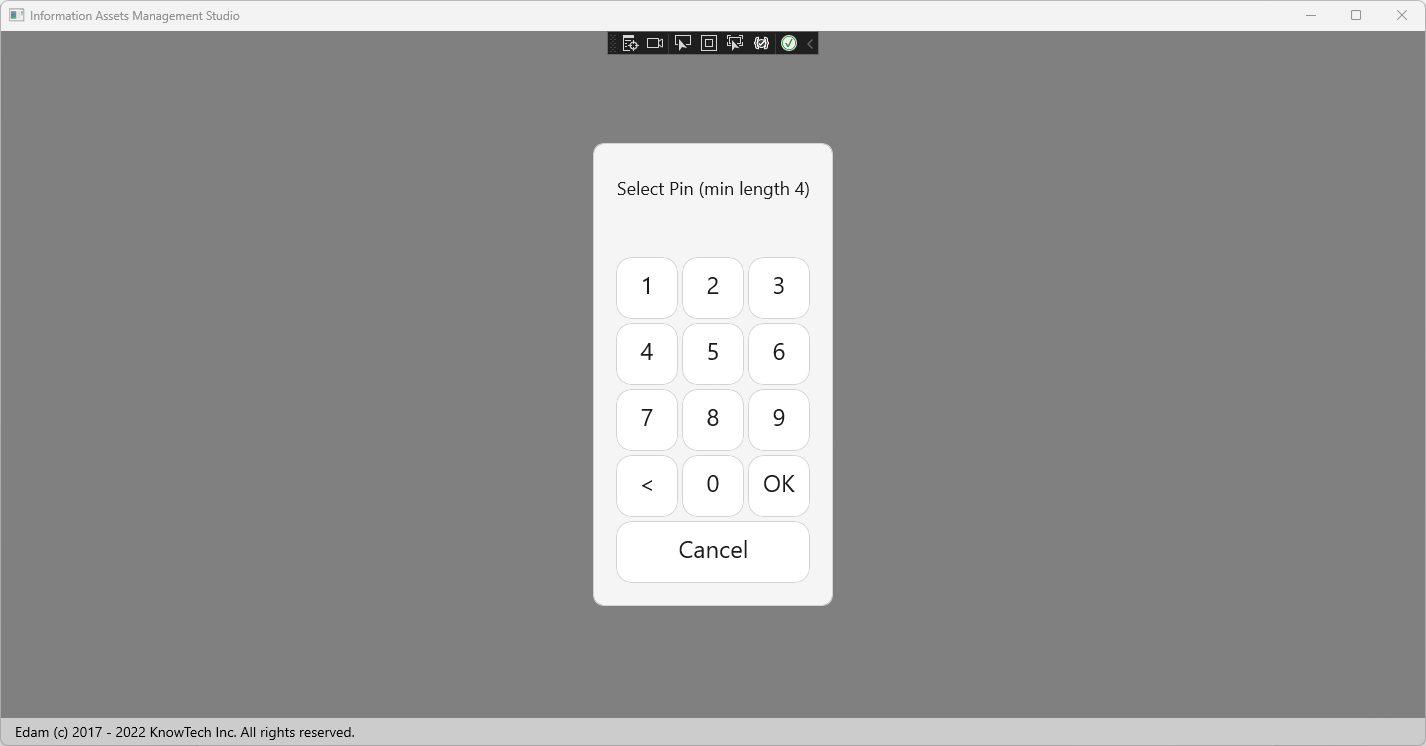
# Running the App for the First Time

Properly configured as stated in the previous section the first time the App is run it will prompt for some basic identity information as shown in the next Figure:

Graphical user interface

Description automatically generated

Some sample values are provided. Understand that the “Agency ID” or “Organization ID” should be a single word not greater than 20 characters to identify the “Tenant” organization. Since the application has been configured as Local the above information is encrypted and stored somewhere and to help retrieve those values next time a Pin number is required and for this reason the next screen follows:



Enter a Pin number that will be requested next time the App is run.

To continue review other documentation provided in the “Documents” folder.