# How to use and compile the sample export extensions

The below few note provide some general help and orientation. It is highly recommended that you read them before starting to work with the code.

# General

The sample export extensions are:

* CMIS -> Exports into CMIS repositories
* SPO -> Exports into SharePoint online (Office 365) and SharePoint 2016 and SharePoint 2013
* SQL -> Exports into Microsoft SQL repositories
* Email -> Send document and data as email
* HelloWorld -> Exports into the file system

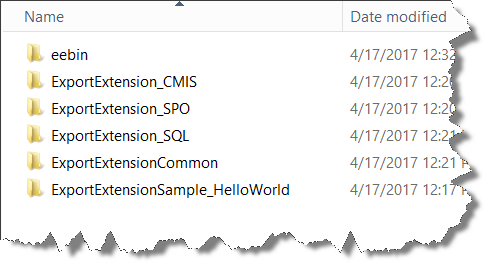
There is one Visual Studio for each of the extensions.

All sample export extensions use a common set of function collected in the ExportExtensionsCommon solution. Therefore, you should first compile ExportExtensionsCommon DLL and then whatever specific export connector you are interested in.

The solutions assume that you have an OCC installation locally on your system at the standard location. That means, the installation folder is assumed to be:

**C:\Program Files (x86)\OpenText\Capture Center 16.2\**

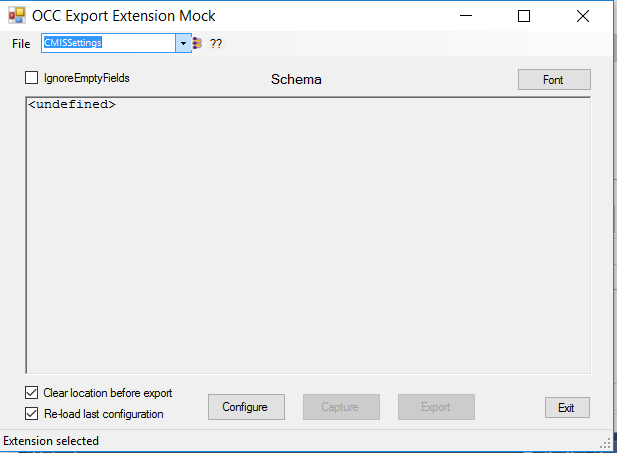
You should copy all samples under a common root folder. When the solutions are compiled, a post build event will copy the DLLs copied into a subfolder under that root; its name is “eebin”. Your structure would look like this:



To make the extensions available in OCC you need to copy the contents of the eebin folder into OCC’s installation folder.

# ExportExtensionCommon

The commonly used DLLs are ExportExtensionCommon.Base.dll and ExportExtensionCommon.WriterBase.dll. Besides these there is a tool called TestAppSIEE.exe that can be used to test the export extension outside of OCC.



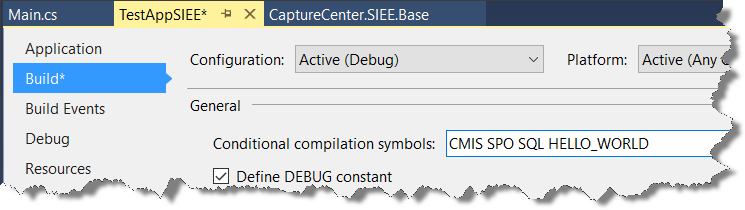
The main action with the tool triggered with the three buttons on the bottom:

* Configure: calls the configuration UI
* Capture: Launches a simple data entry UI
* Export: Exports the document

If “Re-load las configuration” is off an empty configuration UI will be presented, otherwise the last used setting will show up. The capture dialog allows to add a document file either by browsing or using a preselected one. The preselection is available under File -> Preferences. Here you can also add a Default-Values files that contains values that are filled into the form.

It is tedious to repetitively enter the same configuration data (server name, user name, password, etc.). If you are in a configuration dialog for an export extension you may hit <ALT> + F in order to fill in predefined values. These values are taken from a configuration file that you can set when you hit <CRTL> + <ALT> + F. Examples for the configuration files and a sample PDF can be found in ExportExensionCommon/Data.

The tool provides a selector on the top to select one of the export extensions. Which one is available depends on the compilation setting, which are empty by default, to prevent error messages. Set the conditional compilation symbols as needed.



# Export Extensions

All solutions for the export extension samples have the projects:

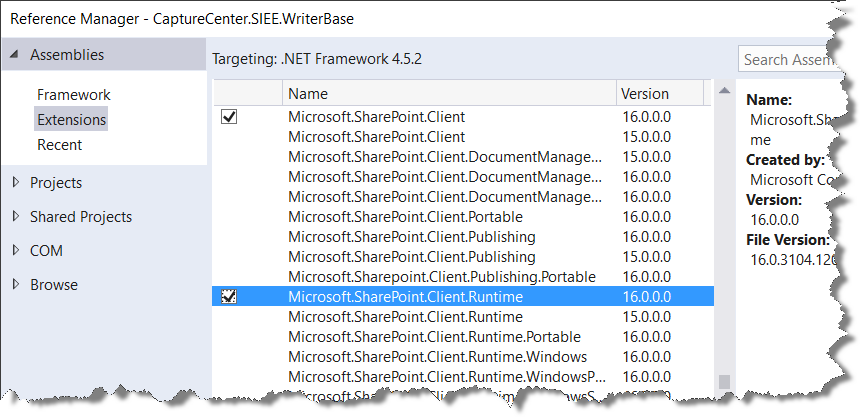
* Adapter: Contains all code implementing the adapter
* Writer: Just one class to register the adapter with OCC
* Client: This project contains the code that talks to the target system
* TestAdapter: Unit tests to test the adapter; they should run after successful compilation
* TestClient: Unit test for the client code. Inside these tests you have a definition for the test systems to test with. These settings must be adapted before the tests can run.

## CMIS connector

The CMIS connector uses the PortCMIS library to talk to CMIS. The sources are included and the compiled DLL must be deployed too.

## SharePoint connector

The SharePoint connector used the CSOM (client side object model) library to talk to SharePoint. It is included as reference but may not be available on your system. You need to install it and can then find the assembly under Assemblies -> Extensions.



## SQL connector

The client unit tests for the SQL connector create test tables in the specified data base.

## Email connector

Nothing special.

## Hello World connector

This connector serves as a demonstrational object and as a jump start when creating new connectors.