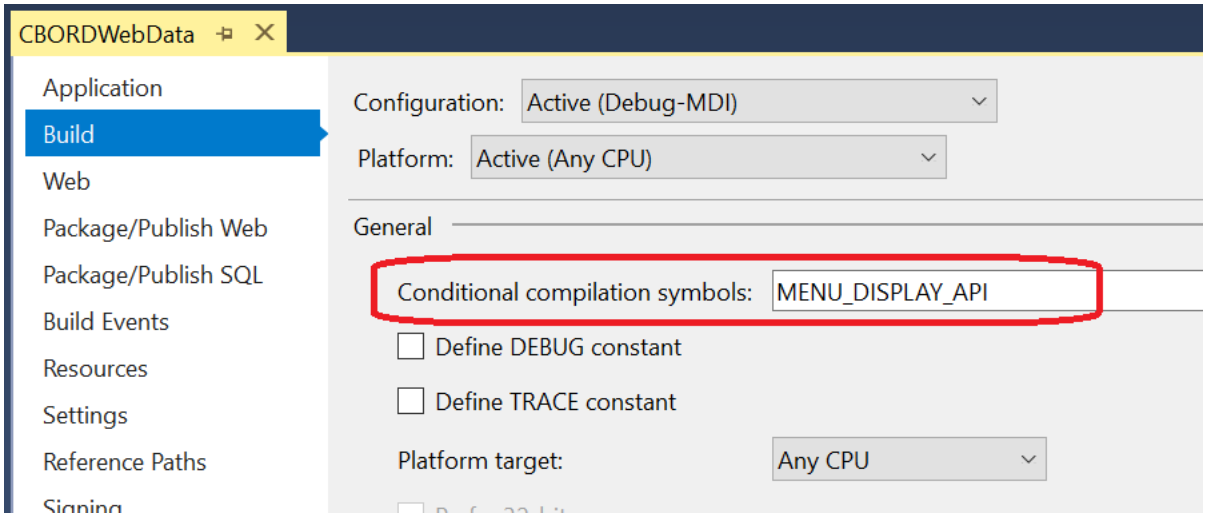
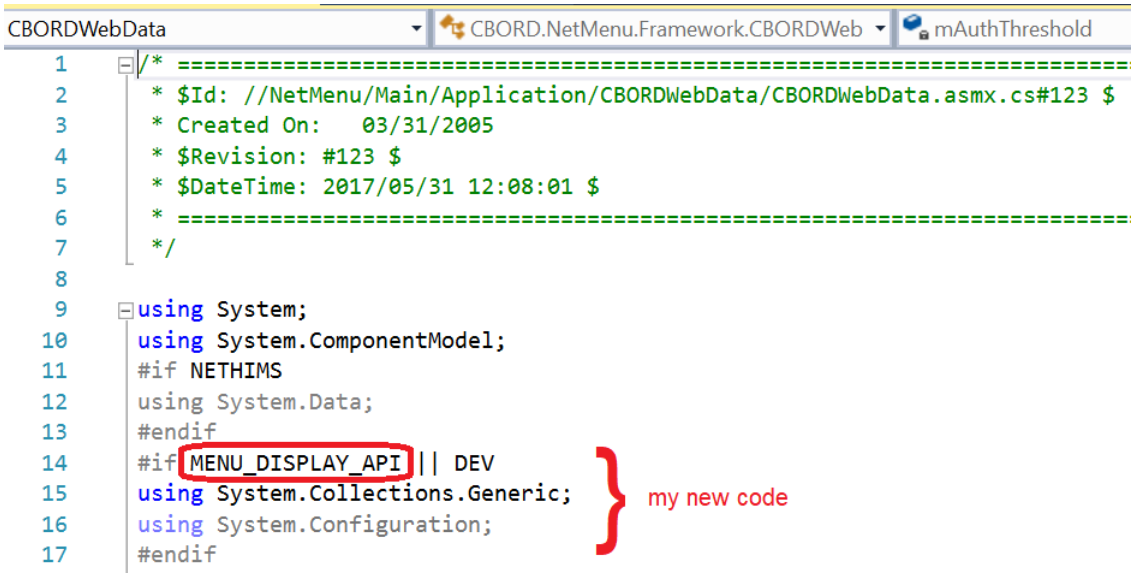
Claire Streb – API Code and Documentation Sample (2017)

I was asked to create a new product, the Menu Display API, using an existing SOAP API project (CBORDWebData).

The first thing I did was to add a new **conditional compilation symbol** to the project so I could compile it without impacting anyone.



Then, I added new code to the project using a **preprocessor directive** that told the compiler to only include my code if the symbol was specified during the build.



I also wanted to automatically generate the API documentation directly from the code using Sandcastle. The following shows how I moved the existing introduction section into a separate file (*include.Default.xml*) and added my new file (*include.MDI.xml*).

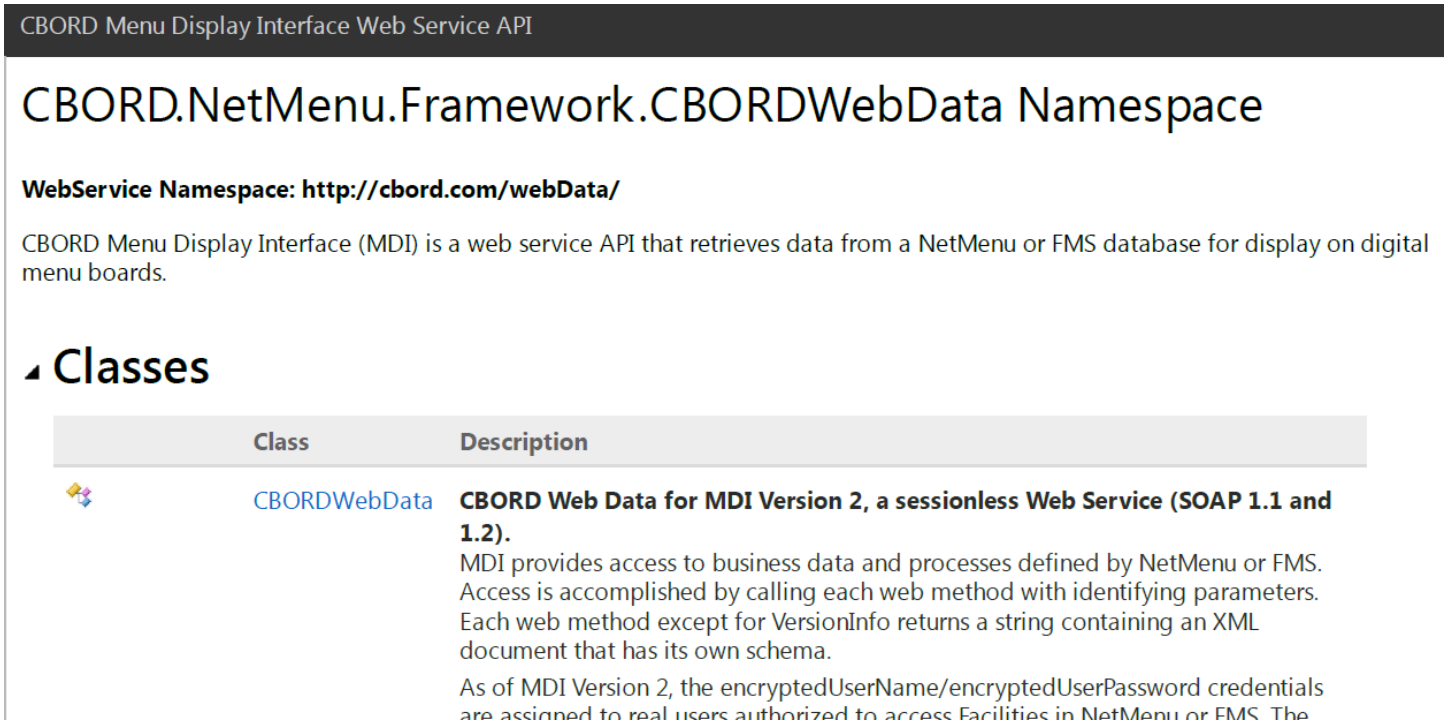
```
34  namespace CBORD.NetMenu.Framework.CBORDWebData
35  {
36  #if DEV || CBORD_INTERNAL || EXTERNAL || NETHIMS
37      /// <summary>
38      /// <include file='include.Default.xml' path='CBORDWebData/CBORDMembers[@name="Name
39      /// </summary>
40  #elif MENU_DISPLAY_API
41      /// <summary>
42      /// <include file='include.MDI.xml' path='CBORDWebData/CBORDMembers[@name="Namespac
43      /// </summary>
44  #else
45      /// <summary>
46      /// CBORDWebData API web methods
47      /// </summary>
48  #endif
```

Here is an excerpt from *include.MDI.xml*:

include.MDI.xml | CBORDWebData.asmx.cs | CBORDWebData

```
1   <!-- include file for MENU_DISPLAY_API -->
2
3   <CBORDWebData>
4
5   <CBORDMembers name="Namespace">
6   <summary>
7   <b>CBORD Web Data for MDI Version 2, a sessionless Web Service (SOAP 1.1 and 1.2).</b>
8   <br/>
9   <para>
10  MDI provides access to business data and processes defined by NetMenu or FMS.  Access is a
11  Each web method except for VersionInfo returns a string containing an XML document that ha
12  </para>
13  <para>
14  As of MDI Version 2, the encryptedUserName/encryptedUserPassword credentials are assigned
```

Here is an excerpt from the auto-generated API documentation based on the *include.MDI.xml* file shown above.



CBORD Menu Display Interface Web Service API

CBORD.NetMenu.Framework.CBORDWebData Namespace

WebService Namespace: http://cbord.com/webData/

CBORD Menu Display Interface (MDI) is a web service API that retrieves data from a NetMenu or FMS database for display on digital menu boards.

Classes

Class	Description
CBORDWebData	**CBORD Web Data for MDI Version 2, a sessionless Web Service (SOAP 1.1 and 1.2).** MDI provides access to business data and processes defined by NetMenu or FMS. Access is accomplished by calling each web method with identifying parameters. Each web method except for VersionInfo returns a string containing an XML document that has its own schema. As of MDI Version 2, the encryptedUserName/encryptedUserPassword credentials are assigned to real users authorized to access Facilities in NetMenu or FMS. The

I added the majority of the new code at the end of the class. Here is a high-level view of the code:

```
2313        #region Menu Display Interface (MDI)
2314    #if MENU_DISPLAY_API || DEV
2315
2316        Public MDI Methods
3198
3199        Private MDI Methods
3390
3391    #endif // MENU_DISPLAY_API || DEV
3392        #endregion //Menu Display Interface (MDI)
3393
3394    }
3395 }
3396
```

Here is an excerpt from expanding line 2316:

```
2313        #region Menu Display Interface (MDI)
2314    #if MENU_DISPLAY_API || DEV
2315
2316        #region Public MDI Methods
2317
2318        /// <summary>
2319        ///  Menu Display Method.  Retrieves zero or more Facilities the user is all
2320        ///</summary>
2321        /// <param name="encryptedUserName">Contact the MDI Administrator for the en
2322        /// <param name="encryptedPassword">Contact the MDI Administrator for the en
2323        /// <param name="facilityExternalId">External ID for a Facility.</param>
2324        /// <param name="getSingleFacility">True to get only a single facility.  Fal
2325        /// <returns>An XML document representing one or more Facilities using the f
2326        /// <code lang="xml" title="MDI Facilities Schema" xml:space="preserve">
2327        /// <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualif
2328        ///   <xs:element name = "MDIFacilities" >
2329        ///     <xs:complexType>
```

Here is a sample of an entire public method I wrote from scratch. Line 2378 is where I call the backend server to return the API method's result (line 2382).

```
2318 /// <summary>
2319 ///  Menu Display Method.  Retrieves zero or more Facilities the user is allowed to access.
2320 ///</summary>
2321 /// <param name="encryptedUserName">Contact the MDI Administrator for the encryptedUserName.</param>
2322 /// <param name="encryptedPassword">Contact the MDI Administrator for the encryptedPassword.</param>
2323 /// <param name="facilityExternalId">External ID for a Facility.</param>
2324 /// <param name="getSingleFacility">True to get only a single facility.  False to get all facilities the user has access to starting
with the specified facility.</param>
2325 /// <returns>An XML document representing one or more Facilities using the following schema:
2326 /// <code lang="xml" title="MDI Facilities Schema" xml:space="preserve">
2327 /// <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
2328 ///   <xs:element name = "MDIFacilities" >
2329 ///     <xs:complexType>
. . .
2355 /// </code>
2356 /// If the ExternalId is empty, it needs to be set in order to use it in MDI.  If the OwningFacilityExternalId is empty, it could
mean: 1) It needs to be set; or
2357 /// 2) The user does not have access to it; or 3) It is the root Facility.
```

```
2358 /// </remarks>
2359 [WebMethod]
2360 public string GetFacilities(string encryptedUserName, string encryptedPassword, string facilityExternalId, bool getSingleFacility)
2361 {
2362     if (string.IsNullOrEmpty(facilityExternalId))
2363     {
2364         var errorMsg = new CBORDException("The facilityExternalId parameter must be specified.");
2365         var exceptionXml = FwUIBL.WriteExceptionToLog(errorMsg, LogFile.GlobalInstance, MethodBase.GetCurrentMethod().Name);
2366         return exceptionXml.OuterXml;
2367     }
2368     int facilityOid;               //object id of the first returned facility
2369     DataServiceConnection ds;      //connection to the backend server
2370     List<int> bizUnitOidList;      //list of business unit object ids
2371     string errorXml;               //returned error from the server
2372     var sc = GetMdiSession(encryptedUserName, encryptedPassword, ref facilityExternalId, out facilityOid, out ds, out
bizUnitOidList, out errorXml);
2373     if (!string.IsNullOrEmpty(errorXml))
2374         return ExtractFirstException(errorXml);
2375     string resultXml;
2376     using (ds)
2377     {
2378         resultXml = MenuDisplay.GetFacilitiesXml(sc, ds.Service, LogFile.GlobalInstance, facilityOid, getSingleFacility,
bizUnitOidList, out errorXml);
2379         if (!string.IsNullOrEmpty(errorXml))
2380             return ExtractFirstException(errorXml);
2381     }
2382     return resultXml;
2383 }
```

The following pages show the above method's resulting API documentation that was auto-generated from the XML comments I wrote above.

CBORDWebDataGetFacilities Method

Menu Display Method. Retrieves zero or more Facilities the user is allowed to access.

Namespace: CBORD.NetMenu.Framework.CBORDWebData
Assembly: CbordWebData (in CbordWebData.dll) Version: 12.14.100.8489

Syntax

C# VB C++ F#

Copy

```
public string GetFacilities(
        string encryptedUserName,
        string encryptedPassword,
        string facilityExternalId,
        bool getSingleFacility
)
```

Parameters

encryptedUserName
Type: SystemString
Contact the MDI Administrator for the encryptedUserName.

encryptedPassword
Type: SystemString
Contact the MDI Administrator for the encryptedPassword.

facilityExternalId

Type: SystemString

External ID for a Facility.

getSingleFacility

Type: SystemBoolean

True to get only a single facility. False to get all facilities the user has access to starting with the specified facility.

Return Value

Type: String

An XML document representing one or more Facilities using the following schema:

MDI Facilities Schema

Copy

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="unqualified" elementFormDef
  <xs:element name="MDIFacilities">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="MDIFacility" minOccurs="1" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Name" type="xs:string" />
              <xs:element name="ExternalId" type="xs:string" />
              <xs:element name="OwningExternalId" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Remarks

Sample XML Result Excerpt Copy

```
<MDIFacilities>
  <MDIFacility>
    <Name>CBORD Master University</Name>
    <ExternalId>Fac1</ExternalId>
    <OwningFacilityExternalId>Fac0</OwningFacilityExternalId>
  </MDIFacility>
</MDIFacilities>
```

If the ExternalId is empty, it needs to be set in order to use it in MDI. If the OwningFacilityExternalId is empty, it could mean: 1) It needs to be set; or 2) The user does not have access to it; or 3) It is the root Facility.

See Also