

MWCIA WCUNDERWRITING Web Service Implementation Guide

Table of Contents

1.	Introduction	2
1.1.	Overview	2
1.2.	Audience	2
2.	Authentication	2
2.1.	Obtaining Authentication Keys	2
2.2.	Address	2
3.	Request Parameters.....	2
4.	Response	3
4.1.	Response Format	3
4.2.	Response Codes and Error Messages	5
5.	Examples	5
5.1.	Example Request.....	5
5.2.	Example Response	6
5.3.	Example Code.....	7
5.3.1.	C# Example Code.....	7
6.	Contact Information.....	8

1. Introduction

1.1. Overview

The MWCIA WCUNDERWRITING Web Service offers automated (system to system) retrieval of Minnesota experience modification history (rating) information. Interstate ratings are not provided by this service. Only final ratings are retrieved – preliminary or contingent ratings are not provided.

Using the WCUNDERWRITING Web Service, up to five years of Minnesota rating information for a specific FEIN or a specific Coverage ID (Risk ID) is retrieved and returned in WCIO WCUNDERWRITING XML format. Please refer to the WCIO website for the WCUNDERWRITING XML schema. Interstate rating information is not returned.

1.2. Audience

This Implementation Guide is intended for insurance carrier software developers who are interested in creating a client application to interact with the MWCIA WCUNDERWRITING product.

2. Authentication

2.1. Obtaining Authentication Keys

In order to connect to the MWCIA WCUNDERWRITING Web Service, both a Web Services User Authentication key and a Product Authentication key are required. These keys can be obtained by your Web Membership Group Administrator via MWCIA's Web Membership system. Please see the Web Membership User Guide for more information on Web Services setup.

2.2. Address

The base address for the WCUNDERWRITING Web Service call is:

<https://www.mwcia.org/WCUnderWriting/ExperienceMod/api/ExperienceModData>

3. Request Parameters

In addition to the User Authentication and Product Authentication keys, the WCUNDERWRITING request must include the following information:

Request Parameter	Description	Parameter Validation
StateCode	22	The state code 22 must always be included.
RiskIdNumber	Coverage ID number	Must be numeric. Either the Coverage ID or FEIN should be included in the request, but not both.

Request Parameter	Description	Parameter Validation
FederalEmployerIdentificationNumber	FEIN	Must be numeric. Either the Coverage ID or FEIN should be included in the request, but not both. Must be 9 digit number.

4. Response

4.1. Response Format

The response will be in WCIO WCUNDERWRITING XML format and will include the following data elements, based on the Coverage ID or FEIN included in the request:

Response Data Element	Description	Notes
Name		
NameReference	Unique reference for each name	Unique reference code for the named insured.
NameOfInsured	Insured name	This is the named insured for the Coverage ID/FEIN. Multiple names are listed in alphabetical order.
FederalEmployerIdentificationNumber	FEIN corresponding to the insured	FEIN corresponding to the named insured.
Address		
AddressReference	Unique reference for each address	Unique address reference code, consisting of the reference code for the corresponding insured name and a sequence number used to identify each address per name.
Street1	Insured address	This is the insured's address/location. Multiple addresses are listed in alphabetical order by state, and then by city.
City	Insured address	See above.
State	Insured address	See above.
ZipCode	Insured address	See above.
ResponseCode	0 = Successful 1 = Data not found 2 = Error in parameters	

Response Data Element	Description	Notes
ResponseMessage	Refer to Response Codes and Error Messages below.	If Response Code = 0, then Response Message 'Successful' is included. If Response Code = 1 or 2, then an error message describing the problem is included. Refer to Response Codes and Error Messages below for more information.
StateCode	22	The state code will always be 22.
RiskIdNumber	Coverage ID	If the request was by Coverage ID, this will be the Coverage ID included in the request. If the request was by FEIN, this will be the Coverage ID corresponding to the FEIN included in the request.
FederalEmployerIdentificationNumber	FEIN	If the request was by FEIN, this will be the FEIN included in the request.
Underwriting Information		
RatingEffectiveDate	Rating effective date, in YYYY-MM-DD format	This is the most current experience modification rating effective date. Multiple rating effective dates and their corresponding ratings will be listed in descending value by Rating Effective Date. Ratings are only available for the last five years. Interstate experience ratings are not returned.
RatingIssueDate	Rating issue date, in YYYY-MM-DD format	
RatingTypeCode	D = Does Not Qualify E = Experience Rating M = Merit Adjustment N = No Current Coverage W = Withdrawn	
ExperienceModificationFactor	Experience modification factor, in decimal form	If there is an experience modification for this rating effective date (Rating Type Code = E or W), this element will be included.
MeritRatingFactor	Merit rating, in decimal form	If there is a merit rating for this rating effective date (Rating Type Code = M or W), this element will be included.
ConstructionClassificationPremiumCreditPercentage	MCPAP factor, in decimal form	If there is a MCPAP (Minnesota Contractors Premium Adjustment Program) factor for this rating effective date, this element will be included.

4.2. Response Codes and Error Messages

If rating data is not found or a problem exists with the request parameters, a Response Code of 1 (Data not found) or 2 (Error in parameters) will be included in the response, along with a specific Response Message, as detailed below.

Response Code	Response Message	Notes
1	No match found for Risk ID number/FEIN	The Risk ID Number/FEIN submitted does not match any Coverage ID in MWCIA systems in the last five years.
1	No rating data exists.	No Minnesota rating data exists within the last five years for the Coverage ID or FEIN entered.
2	Not a valid security key.	Authentication key is not valid
2	Request must include state code 22.	The state code is null or incorrect.
2	Request must include either Risk ID or FEIN.	The Risk ID Number and/or the FEIN are null.
2	Please provide Risk ID Number in correct format (numeric up to 7 digits).	Coverage ID must be numeric with up to 7 digits.
2	Please provide FEIN in correct format (numeric 9 digits).	FEIN must be numeric with exactly 9 digits.
2	Schema violation.	Error in the schema.
2	System error.	Error in the system.
2	Multiple matches found for FEIN.	If the FEIN matches multiple Coverage IDs in MWCIA systems in the last five years and the most recent cannot be determined, no results will be returned.

5. Examples

Following are examples of web service request calls to search by coverage ID (Risk ID) or to search by FEIN.

<https://www.mwcia.org/WCUnderWriting/ExperienceMod/api/ExperienceModData?coverageID=1234567&statecode=22>

<https://www.mwcia.org/WCUnderWriting/ExperienceMod/api/ExperienceModData?FEIN=123456789&statecode=22>

5.1. Example Request

GET

<https://www.mwcia.org/WCUnderWriting/ExperienceMod/api/ExperienceModData?coverageID=1234567&statecode=22>

Authorization: Basic

QlFBQUFBPT06ZWpwcDJKMWZwd2Q3ZmVnNm1waUdaUXR4UG8rUTk1NGo4K1lTUkRZTU5HQQT0=

5.2. Example Response

```
<?xml version="1.0" encoding="utf-8"?>
<Submission xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://www.wcio.org/XML/WCUNDERWRITING">
  <WCUnderwriting>
    <Names>
      <Name nameReference="001">
        <NameOfInsured>INSURED NAME</NameOfInsured>
        <FederalEmployerIdentificationNumber>123456789</FederalEmployerIdentificationNumber>
      </Name>
    </Names>
    <Addresses>
      <Address addressReference="001-1">
        <Street1>123 OAK DR</Street1>
        <City>A CITY</City>
        <State>MN</State>
        <ZipCode>555559999</ZipCode>
      </Address>
      <Address addressReference="001-2">
        <Street1>321 MAIN ST</Street1>
        <City>ANOTHER CITY</City>
        <State>MN</State>
        <ZipCode>555519999</ZipCode>
      </Address>
    </Addresses>
    <ResponseCode>0</ResponseCode>
    <ResponseMessage>Successful</ResponseMessage>
    <StateCode>22</StateCode>
    <FederalEmployerIdentificationNumber>123456789</FederalEmployerIdentificationNumber>
    <UnderwritingsInformation>
      <UnderwritingInformation>
        <RatingEffectiveDate>2018-05-11</RatingEffectiveDate>
        <RatingIssueDate>2017-12-22</RatingIssueDate>
        <RatingTypeCode>E</RatingTypeCode>
        <ExperienceModificationFactor>0.91</ExperienceModificationFactor>
      </UnderwritingInformation>
      <UnderwritingInformation>
        <RatingEffectiveDate>2017-05-11</RatingEffectiveDate>
        <RatingIssueDate>2016-12-20</RatingIssueDate>
        <RatingTypeCode>E</RatingTypeCode>
        <ExperienceModificationFactor>0.92</ExperienceModificationFactor>
      </UnderwritingInformation>
      <UnderwritingInformation>
        <RatingEffectiveDate>2015-09-30</RatingEffectiveDate>
        <RatingIssueDate>2015-06-08</RatingIssueDate>
        <RatingTypeCode>W</RatingTypeCode>
        <ExperienceModificationFactor>0.91</ExperienceModificationFactor>
      </UnderwritingInformation>
    </UnderwritingsInformation>
  </WCUnderwriting>
</Submission>
```

```

        <ConstructionClassificationPremiumCreditPercentage>0.9</Construction
ClassificationPremiumCreditPercentage>
    </UnderwritingInformation>
</UnderwritingsInformation>
</WCUnderwriting>
</Submission>"

```

5.3. Example Code

5.3.1. C# Example Code

The example code below targets .Net Framework 4.5.2.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net.Http;
using System.Net.Http.Headers;

namespace HttpClientSample
{
    class HttpClientService
    {
        static HttpClient client = new HttpClient();
        static void Main(string[] args)
        {
            RunAsync().GetAwaiter().GetResult();
        }

        static async Task<String> GetXMLAsync(string path)
        {
            String xmlData = null;
            HttpResponseMessage response = await client.GetAsync(path);

            if (response.IsSuccessStatusCode)
            {
                //response body gets the WCUNDERWRITING XML.
                xmlData = await response.Content.ReadAsStringAsync();
                // Now, you have XML data and you can process it.
            }
            else
            {
                //Some error happened and now return the error message
                return await response.Content.ReadAsStringAsync();
            }
            return xmlData;
        }

        static void ShowXML(String xml)
        {
            // Write out the data to console so you can see what is returned from the web
            // service.

```

```

        Console.WriteLine(xml);
    }

    static async Task RunAsync()
    {
        string userAuthKey      = "User Authentication Key from Web Membership
System";
        string productAuthKey   = "Product Authentication Key from Web
Membership System";
        string baseAddress      =
"https://www.mwcia.org/WCUnderWriting/ExperienceMod/";
        string requestUri       =
"api/ExperienceModData?coverageID=1951009&statecode=22";

        // Setup the URL for the service
        client.BaseAddress = new Uri(baseAddress);
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/xml"));

        //Now, put together the userAuthKey and productAuthKey as basic
authentication and add it to request header.
        byte[] credentials = Encoding.ASCII.GetBytes(userAuthKey + ":" +
productAuthKey);
        client.DefaultRequestHeaders.Authorization = new
AuthenticationHeaderValue("Basic", Convert.ToBase64String(credentials));
        try
        {

            // Get the Mod Data in WCUNDERWRITING XML format
            // String XML = HttpUtility.HtmlDecode(await GetXMLAsync(requestUri));
            String XML = await GetXMLAsync(requestUri);

            ShowXML(XML);

        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }

        Console.ReadLine();
    }
}
}

```

6. Contact Information

If you have questions or concerns not addressed by this guide, please email WebMembership@mwcia.org or call the MWCIA at 952-897-1737.