

# Mondor Software

Currency Exchange Rates Web Service

version 1.4

## API Manual

### Definition

Currency Exchange Rates Web Service (CERWS) is an XML web service which provides currency exchange rates to its clients. Rates are updated every 20 minutes.

Clients may retrieve currency list (which is constant), rates for various currencies, or automatically convert one currency into another. Automatic conversion can be used to outsource the actual conversion, as this is counted as only one call to web service (see Licensing below).

### Licensing

Every client must have a license key, which he submits with every request. Every request to the web service, like currency rate query or conversion, is counted towards clients' license key, except if the license key is not limited to the amount of calls. License keys may be limited by amount of calls that could be made or by date in which clients' calls will be processed.

There is a special functionality that allows client to retrieve the current balance (both calls and days left) and update the existing key for a special price. So you don't have to change the key ID to extend its' functionality – simply get the new update key (it's cheaper than the new one) and update it using either a web service function or a web page.

When your key expires, you're starting getting zero results – i.e. the result of currency exchange or currency rate is 0. Some keys allows you to setup SMS reminder when your key is about to expire (you can set it to send you SMS when calls limit reaches certain level or expiration date is near).

### Use

You can incorporate CERWS into your web application, as well as into any application which may consume XML Web Services. Probably the easiest way to do that is using Microsoft Visual Studio. You can download free version of latest Microsoft Visual Studio at [Microsoft download website](#).

Note though, that you can use XML Web Services also from other popular environments like PHP, which is out of scope of this document.

You can see the CERWS in action using the downloadable sample project (for Visual Studio 2008).

1. Acquire your access key. Either buy or request a demo key from [www.mondor.org](http://www.mondor.org).
2. Create or open existing project. Click on the project name and either click on “Add service reference” (Visual Studio 2008) or “Add web service reference” (later versions of Visual Studio).
3. Enter **<http://mondor.org/ces/rates.asmx>** as the path to the web service, if you prefer using SOAP-based XML Web Services.  
Enter **<http://mondor.org/ces/rates.svc>** as the path to the web service, if you prefer using WCF Web Services.  
For WebAPI reference, see below.
4. Now you can use the CERWS object inside of your application (see examples below)

### Example 1: Retrieving the list of currencies.

In this example we will fill the drop down list web control with the list of all available currencies.

```
//Load combo boxes with currency names
cews.rates crates = new CERWSClient.cews.rates();
string[] currency = crates.GetCurrencyCodes();
foreach (string cname in currency)
{
    ddlSource.Items.Add(cname);
    ddlTarget.Items.Add(cname);
}
```

As you can see, first we instantiated an object from the web service. Then, we created an array of strings and filled it using a function of the web service – **GetCurrencyCodes**. Then, we simply filled the drop down list from that array of strings.

### Example 2: Converting currency.

You can find this example in the attached source code (see [www.mondor.org](http://www.mondor.org) for a source code). We’re having a form with a drop down lists (**ddlSource**, **ddlTarget**) which contains the list of currencies taken from the example above, command button named **cmdConvert**, and 2 text fields – **txtAcode**, which contains your access code and **txtSource** which contains the amount you wish to convert (e.g. 5). When you type your amount, select the source and target currencies, then push the button, result is taken from the web service and printed into the resulting **txtTarget** text field.

```
private void cmdConvert_Click(object sender, EventArgs e)
{
    //Check access code
    string Acode = txtAcode.Text.Trim();
    if (Acode.Length != 32)
    {
        //For testing purposes, the code can be acquired free of charge.
        MessageBox.Show("Invalid access code! Please acquire new one at
www.mondor.org or fill the existing code!");
        return;
    }
    //Connect to web services (see URL in app.config)
```

```

        cews.rates crates = new CERWSClient.cews.rates();
        //Get the conversion results - just one line of code
        txtTarget.Text = crates.Convert(ddlSource.Text, ddlTarget.Text,
Convert.ToDouble(txtSource.Text), ACode).ToString();
        //Free resources used
        crates.Dispose();
    }

```

### Example 3 – Retrieving the current balance (calls or days left)

```

//Connect to web services (see URL in app.config)
cews.rates crates = new CERWSClient.cews.rates();
int CurrentBalance = crates.CheckBalance(txtACode.Text);

```

### Example 4 – Upgrading one access code with another

```

string OriginalCode = txtACode.Text;
string UpgradeCode = txtUpCode.Text;
cews.rates crates = new CERWSClient.cews.rates();
if (crates.UpgradeAccessCode(OriginalCode, UpgradeCode))
{
    MessageBox.Show("Code upgraded successfully!");
}
else
{
    MessageBox.Show("For some reasons, upgrade failed!");
}

```

### Example 5 – Invoking from PHP

```

<?php

// connect to the server
$soap_client = new SoapClient('http://www.mondor.org/ces/rates.asmx?WSDL');

// convert from us dollar to euro
$conversion = $soap_client->Convert(array(
    'CurrencyFrom' => 'USD',
    'CurrencyTo' => 'EUR',
    'ValueFrom' => (double)1,
    'UserKey' => 'YOUR_USER_KEY',
));

echo "Result: {$conversion->ConvertResult}";

?>

```

## Functions available in XML Web Services / WCF Services

### CheckBalance

This function returns the remaining calls for your access code. This function is free, it doesn't cost you a credit. If your access code is limited by date, not by amount of calls, the returning value is irrelevant and will be equal to 5.

### **CheckExpirationDate**

The same as before, but returns the expiration date for your access code.

### **Convert**

This is the major function which performs the conversion. Each call costs you one credit if your access code is limited by amount of calls.

### **ConvertByDate**

The same as before, but you are able to set the date of conversion. If available, the rate of that day will be used for conversion. If the rate for that date is unavailable, you'll get zero return value.

### **GetCurrencyCodes**

This function returns the list of codes for all available currencies.

### **GetCurrencyList**

This function returns the list of all available currencies.

### **GetMinimalDate**

If you want to use the ConvertByDate function, this function returns the minimal date for which such conversion is possible. It is not guaranteed, though, to have data for This function is free.

## **WebAPI**

If you are using WebAPI, functions have a bit different names and parameters. For examples below, let's imagine that your API key is E7CC7BD10D414DC6999905E4CE049952. In that case, function calls would look like this:

### **Get currency list**

You can retrieve the list of supported currencies by calling the `getcurrencylistcsv` method. It returns the list of currency in comma-separated-value format, 1 currency per line.

<https://mondor.org/convert/E7CC7BD10D414DC6999905E4CE049952/getcurrencylistcsv>

Returns:

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">
```

```
<script/>
```

AED;UAE Dirham ALL;Albanian Lek AMD;Armenian Dram ARS;Argentine Peso  
AUD;Australian Dollar AWG;Aruba Florin AZN;Azerbaijani Manat BBD;Barbados Dollar  
BDT;Bangladesh Taka BGN;Bulgarian Lev BHD;Bahraini Dinar BIF;Burundi Franc  
BMD;Bermuda Dollar BND;Brunei Dollar BOB;Bolivian Boliviano BRL;Brazilian Real  
BSD;Bahamian Dollar BTN;Bhutan Ngultrum BWP;Botswana Pula BYN;Belarusian Ruble  
(new) BYR;Belarus Ruble BZD;Belize Dollar CAD;Canadian Dollar CHF;Swiss Franc  
CLP;Chilean Peso CNY;Chinese Yuan COP;Colombian Peso CRC;Costa Rica Colon  
CZK;Czech Koruna DKK;Danish Krone DOP;Dominican Peso DZD;Algerian Dinar  
EGP;Egyptian Pound ETB;Ethiopian Birr EUR;Euro FJD;Fiji Dollar GBP;British Pound  
GEL;Georgian Lari GHS;Ghanian Cedi GMD;Gambian Dalasi GNF;Guinea Franc  
GTQ;Guatemala Quetzal HKD;Hong Kong Dollar HNL;Honduras Lempira HRK;Croatian  
Kuna HTG;Haiti Gourde HUF;Hungarian Forint IDR;Indonesian Rupiah ILS;Israeli  
Shekel INR;Indian Rupee IQD;Iraqi Dinar IRR;Iran Rial ISK;Iceland Krona  
JMD;Jamaican Dollar JOD;Jordanian Dinar JPY;Japanese Yen KES;Kenyan Shilling  
KGS;Kyrgystani Som KMF;Comoros Franc KRW;Korean Won KWD;Kuwaiti Dinar KYD;Cayman  
Islands Dollar KZT;Kazakhstan Tenge LBP;Lebanese Pound LKR;Sri Lanka Rupee  
LSL;Lesotho Loti MAD;Moroccan Dirham MDL;Moldovan Leu MGA;Malagasy Ariary  
MKD;Macedonian Denar MNT;Mongolian Tugrik MOP;Macau Pataca MRO;Mauritania  
Ougulya MUR;Mauritius Rupee MVR;Maldives Rufiyaa MWK;Malawi Kwacha MXN;Mexican  
Peso MYR;Malaysian Ringgit MZN;Mozambican Metical NAD;Namibian Dollar  
NGN;Nigerian Naira NIO;Nicaragua Cordoba NOK;Norwegian Krone NPR;Nepalese Rupee  
NZD;New Zealand Dollar OMR;Omani Rial PAB;Panama Balboa PEN;Peruvian Nuevo Sol  
PGK;Papua New Guinea Kina PHP;Philippine Peso PKR;Pakistani Rupee PLN;Polish  
Zloty PYG;Paraguayan Guarani QAR;Qatar Rial RON;Romanian New Leu RSD;Serbian  
Dinar RUB;Russian Rouble RWF;Rwanda Franc SAR;Saudi Arabian Riyal SBD;Solomon  
Islands Dollar SCR;Seychelles Rupee SEK;Swedish Krona SGD;Singapore Dollar  
SLL;Sierra Leone Leone SVC;El Salvador Colon SZL;Swaziland Lilageni THB;Thai  
Baht TND;Tunisian Dinar TOP;Tonga Paanga TRY;New Turkish Lira TTD;Trinidad  
Tobago Dollar TWD;Taiwan Dollar TZS;Tanzanian Shilling UAH;Ukraine Hryvnia  
UGX;Ugandan Shilling USD;U.S. Dollar UYU;Uruguayan New Peso VEF;Venezuelan  
Bolivar VND;Vietnam Dong VUV;Vanuatu Vatu WST;Samoa Tala XAF;CFA Franc (BEAC)  
XCD;East Caribbean Dollar XOF;CFA Franc (BCEAO) XPF;Pacific Franc YER;Yemen  
Riyal ZAR;South African Rand ZMW;Zambian Kwacha

</string>

## Get API key balance

Balance is number of calls left for your API key. If your key is not limited by number of calls, you will get “-1” as result.

<https://mondor.org/convert/E7CC7BD10D414DC6999905E4CE049952/balance>

Returns:

<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">

```
<script/>
```

```
-1
```

```
</string>
```

When your balance is depleted, the number should be 0, otherwise it's positive integer that represents the number of calls left.

## Get API key expiration date

When your key is limited by date (e.g. Annual or Monthly), you can check the expiration date instead of balance. Some keys may be limited by both balance and date (e.g. Free and Trial keys, or Annual 100, which is limited to 100 calls per day for a year).

<https://mondor.org/convert/E7CC7BD10D414DC6999905E4CE049952/expiration>

Returns:

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">
```

```
<script/>
```

```
Monday, January 01, 0001
```

```
</string>
```

Note, that keys that are not limited by date, will return the minimal date (01/01/01).

## Convert currency

<https://mondor.org/convert/E7CC7BD10D414DC6999905E4CE049952/USD/AUD/3/05082018>

In the example above, USD is the source currency, AUD the target currency, 3 is the amount of source currency, and 05082018 is the date. We are converting 3 U.S. dollars to Australian dollars using the rates of 5<sup>th</sup> of August 2018.

Amount and Date are optional parameters. You can use shorter request:

<https://mondor.org/convert/E7CC7BD10D414DC6999905E4CE049952/USD/AUD>

If you want to retrieve the general rate between two currencies.

Default amount is 1. When the date is omitted, the current rate is used.