

Mondor Software

Currency Exchange Rates Web Service

version 1.1

User 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 Express Edition 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.
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”.
3. Enter **http://www.mondor.org/ces/rates.asmx** as the path to the web service.
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 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}";

?>
```