```
<html>
<head>

<script>
////////////////////////////////////////////////////////////////////////////
// FILE    : RssBoCExchange.htm
// PURPOSE : Prueba de lectura de ficheros Rss Xml del Bank of Canada con
// cambios de divisas.
//
// El sistema consta de una lista de objetos, uno por divisa, cada objeto
// tiene los cambios de esa divisa tanto a euros como a dolares canadienses.
// Cada objeto se inicia con un cambio por defectos y si se le pide un cambio
// aplica esos valores por defecto. Pero una vez cargada la pagina, cada uno
// de los objetos, en secuencia, va pidiendo su cambio actualizado al Bank of
// Canada y si recibe la informacion pone al dia su valor y aplica ese cambio
// a partir de ese momento.
//
// El objeto 0 es el del dolar canadiense, del que no se puede pedir cambio
// pues es la moneda de referencia del Bank of Canada, el objeto 1 es el euro,
// cuando actualiza su cambio a dolares canadienses tambien actualiza el
// objeto dolar canadiense respecto del euro. Cuando el euro ya esta
// actualizado, pide que se haga la actualizacion del siguiente objeto, el
// dolar americano (2) y este al finalizar el siguiente cambio.
//
// Pueden crearse por programacion tantos objetos como divisas se quieran.
// En esta prueba ademas de los enumerados estan la libra esterlina, el franco
// suizo, el yen japones y la corona checa.
//
// Este sistema se ha desarrollado y probado para FireFox e Internet Explorer.
// Dado que por razones de seguridad no se pueden abrir desde javascript
// ficheros Rss de diferentes hosts al que alberga la pagina que contine a
// el codigo javascript, se necesita un puente que residiendo en el mismo host
// que la pagina capture dichos ficheros. Este puente se hace mediante un
// programa Php.
//
// Estas restricciones de seguridad las tiene oficialmente FireFox y otros
// navegadores. En principio, Internet Explorer parece no tenerlas, por lo
// que en las primeras versiones de este programa se establecia que IE fuera
// directamente por la pagina Rss. Sin embargo, esto a veces funcionaba y
// otras no y, como a su vez, suele ser bueno unificar criterios, sea cual
// sea el tipo de navegados, este codigo pide a un programa Php (rssget.php)
// que le traiga el fichero Rss que necesita para cada divisa.
//
// El codigo incluye todas las funciones auxiliares que necesita, el codigo
// html y los estilos css (Cascade Style Sheet) en un mismo fichero para
// facilitar una vision de conjunto. Obviamente el programa Php esta
// codificado en un programa separado y la version operativa real de este
// codigo este descompuesta en sus diferentes modulos.
//
// Este codigo esta basado en el codigo rssfetch.php de:
// - Author: Paul Sobocinski
// - Title: RSS and AJAX: A Simple News Reader
// - Created: September 13, 2006
// - http://www.xml.com/pub/a/2006/09/13/rss-and-ajax-a-simple-news-reader.html
//
// Como los resultados se van mostrando una vez que toda la pagina ya esta
// cargada se ha de realizar inyectando codigo dentro del <div> puestos a
// tal efecto.
////////////////////////////////////////////////////////////////////////////

```

```
61  ////////////////////////////////////////////////////////////////////////////
62  // DATA FUNCTIONS
63  ////////////////////////////////////////////////////////////////////////////
64
65  ////////////////////////////////////////////////////////////////////////////
66     function DatRound(datNum, // Number
67                       decPos) // Number of decimal places
68
69  // PURPOSE: Returns a number rounded to a decPos number of decimal places.
70  ////////////////////////////////////////////////////////////////////////////
71  { return(Math.round(datNum*Math.pow(10,decPos))/Math.pow(10,decPos)); }
72
73
74  ////////////////////////////////////////////////////////////////////////////
75  // TEXT FUNCTIONS
76  ////////////////////////////////////////////////////////////////////////////
77
78  ////////////////////////////////////////////////////////////////////////////
79     function TxtReplace(inpTxt, oldTxt, newTxt)
80
81  // PURPOSE: Returns a text with oldTxt change by newTxt. Does not use
82  //          regular expressions.
83  ////////////////////////////////////////////////////////////////////////////
84  {
85    var tmpArr = inpTxt.split(oldTxt);
86    return(tmpArr.join(newTxt));
87  }
88
89
90  ////////////////////////////////////////////////////////////////////////////
91     function TxtBetween2Tag(inpTxt, // Text
92                             tagIni, // Initial tag
93                             tagEnd) // End tag
94
95  // PURPOSE: The simplest TxtBetween2Tag(), no case sensitive, mandatory and
96  //          no compact. .substring(i,e) stops at position e and does not
97  //          include the character at position e.
98  ////////////////////////////////////////////////////////////////////////////
99  {
100   var txtBet = "";
101   var posIni = inpTxt.indexOf(tagIni);
102   if(posIni >= 0) // If tagIni found
103   {
104     var lenIni = tagIni.length;
105     var posSub = posIni + lenIni;
106     var posEnd = inpTxt.indexOf(tagEnd, posSub);
107     if(posEnd >= posSub) { txtBet = inpTxt.substring(posSub,posEnd); }
108   }
109   return(txtBet);
110 };
111
112
113 ////////////////////////////////////////////////////////////////////////////
114 // HTML FUNCTIONS
115 ////////////////////////////////////////////////////////////////////////////
116
117 ////////////////////////////////////////////////////////////////////////////
118    function HtmClsLtGt(inpHtm)
119
120 // PURPOSE: Returns a html text with < and > change by &lt; and &gt;.
```

```
121 ////////////////////////////////////////////////////////////////////////////
122 {
123   var notLt = TxtReplace(inpHtm, "<", "&lt;");
124   var notGt = TxtReplace(notLt,  ">", "&gt;");
125   return(notGt);
126 }
127
128
129 ////////////////////////////////////////////////////////////////////////////
130    function HtmClsLtGtBold(inpHtm)
131
132 // PURPOSE: Returns a html text with < and > change by &lt; and &gt;, after
133 //          that puts all text (not tags) in bold. This function does not
134 //          work well if there are previous &lt; or &gt;. To correct this
135 //          need previous replacement to preserve the originals &lt; &gt and
136 //          a final replacement to retore originals &lt; &gt.
137 ////////////////////////////////////////////////////////////////////////////
138 {
139   var nLtGt = HtmClsLtGt(inpHtm);
140   var rep01 = TxtReplace(nLtGt,  "&lt;", "</b>&lt;");
141   var rep02 = TxtReplace(rep01,  "&gt;", "&gt;<b>");
142   return(rep02);
143 }
144
145
146 ////////////////////////////////////////////////////////////////////////////
147    function HtmInjectCode(id,     // Elemente identification
148                           htmCod) // Html code
149
150 // PURPOSE: Returns true if can inject the code htmCod in the element id.
151 ////////////////////////////////////////////////////////////////////////////
152 {
153   var htmEle = document.getElementById(id);
154   if(!htmEle) { alert(id);                          return(false); }
155   else        { htmEle.innerHTML = htmCod; return(true);  }
156 }
157
158
159 ////////////////////////////////////////////////////////////////////////////
160 // XML FUNCTIONS
161 ////////////////////////////////////////////////////////////////////////////
162
163 ////////////////////////////////////////////////////////////////////////////
164    function XmlGetValue(itm, atr)
165
166 // PURPOSE: Returns a text value. If atr = cb:value and itm =
167 // ...<cb:value frequency="business" decimals="4">0.9947</cb:value>...
168 // first gets " frequency="business" decimals="4">0.9947<" and returns 0.9947
169 ////////////////////////////////////////////////////////////////////////////
170 {
171   var tmp = TxtBetween2Tag(itm,"<"+atr,"/"+atr+">");
172   var txt = TxtBetween2Tag(tmp,">","<");
173   return(txt);
174 }
175
176
177 ////////////////////////////////////////////////////////////////////////////
178 // BANK OF CANADA OBJECT, FUNCTIONS AND DATABASE
179 //
180 // http://www.bankofcanada.ca/en/rates/exchange-look.html normal exchange
```

```
// http://www.bankofcanada.ca/en/rates/rss_fx.html        all rss exchange
////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////
// CONSTANTS
////////////////////////////////////////////////////////////////////////////
var BoCDecPos =  4;   // Decimal positions
var BoCErrLog = "";   // Log for errors
var BoCTrcLog = "";   // Log for trace
var BoCTrcXml = true; // If true show XML


////////////////////////////////////////////////////////////////////////////
// CONTROL
////////////////////////////////////////////////////////////////////////////
var BoCArr = new Array; // Works as array (access by numbers)
var BoCLst = new Array; // Works as asociative list (access by strings)


////////////////////////////////////////////////////////////////////////////
// BANK OF CANADA ACCESS FUNCTIONS
////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////
   function BoCAddErr(newErr) // Error messaje

// PURPOSE: Store a new error messaje
////////////////////////////////////////////////////////////////////////////
{
  if(BoCErrLog=="") { BoCErrLog = newErr; }
  else              { BoCErrLog = BoCErrLog + "<br>" + newErr; }
}


////////////////////////////////////////////////////////////////////////////
   function BoCAddTrc(newTrc) // Error messaje

// PURPOSE: Store a new trace messaje
////////////////////////////////////////////////////////////////////////////
{
  if(BoCTrcLog=="") { BoCTrcLog = newTrc; }
  else              { BoCTrcLog = BoCTrcLog + " " + newTrc; }

  HtmInjectCode("BoCTrcDiv", BoCTrcLog);
}


////////////////////////////////////////////////////////////////////////////
   function BoCXmlTrc(urlRss, // RSS address
                      xmlTxt) // XML text

// PURPOSE: Show the Xml text only if BoCXmlView is true
////////////////////////////////////////////////////////////////////////////
{
  if(BoCTrcXml)
  {
    var htmTxt = "<h1>"+urlRss+"</h1>"+HtmClsLtGtBold(xmlTxt);
    HtmInjectCode("BoCXmlDiv", htmTxt);
  }
}
```

```
241
242
243 ////////////////////////////////////////////////////////////////////////////
244 // BANK OF CANADA RSS FUNCTIONS
245 ////////////////////////////////////////////////////////////////////////////
246
247 ////////////////////////////////////////////////////////////////////////////
248    function BoCRssGet(bocPos, // Array position begins with 1 for EUR, 2...
249                       urlRss, // Only uses for trace
250                       urlPhp) // Via Php in the same host (Non IE)
251
252 // PURPOSE: Returns the Xml as text.
253 //          Uses an indirect Php via for all browsers.
254 ////////////////////////////////////////////////////////////////////////////
255 {
256   var chkXml = true;
257   if(window.ActiveXObject) { xhr = new ActiveXObject("Microsoft.XMLHTTP"); }
258   else if(window.XMLHttpRequest)              { xhr= new XMLHttpRequest(); }
259   else                                                { chkXml = false; }
260
261   if(chkXml) // If is ok
262   {
263     xhr.open("GET", urlPhp, true); // Prepare the xml http request object
264
265     xhr.setRequestHeader("Cache-Control", "no-cache");
266     xhr.setRequestHeader("Pragma", "no-cache");
267     xhr.onreadystatechange = function() // Execute when ready (asincronous)
268     {
269       if(xhr.readyState == 4)
270       {
271         if(xhr.status == 200)
272         {
273           if(xhr.responseText != null)
274           {
275             BoCXmlTrc(urlRss, xhr.responseText); // Only for show
276             if(bocPos==1) { BoCUpdEurCaD(bocPos, xhr.responseText); }
277             else          { BoCUpdKey   (bocPos, xhr.responseText); }
278           }
279           else
280           { BoCAddErr("["+bocPos +"|"+xhr.status+"|"+xhr.statusText+"];"); }
281         }
282         else
283         { BoCAddErr("["+bocPos +"|"+xhr.status+"|"+xhr.statusText+"];"); }
284       }
285     }
286     xhr.send(null); // Send the request
287   }
288 }
289
290
291 ////////////////////////////////////////////////////////////////////////////
292 // OBJECT FUNCTIONS
293 ////////////////////////////////////////////////////////////////////////////
294
295 ////////////////////////////////////////////////////////////////////////////
296    function BoCObj(bocKey, // Currency code (text)
297                    bocNam, // Currency name
298                    bocRss, // Absolut Rss url at Bank of Canada
299                    bocPhp, // Php at the same host
300                    bocCaD, // 1 CaD = bocCaD bocKey
```

```
                    bocEur, // 1 Eur = bocEur bocKey
                    bocTyp, // noon, close rate or default when not updated
                    bocDte) // Get date

// PURPOSE: Create a currency exchange database object.
////////////////////////////////////////////////////////////////////////////
{
  this.bocKey = bocKey; // Currency code (text)
  this.bocNam = bocNam; // Currency name
  this.bocRss = bocRss; // Absolut Rss url at Bank of Canada
  this.bocPhp = bocPhp; // Php at the same host
  this.bocCaD = bocCaD; // 1 CaD = bocCaD bocKey
  this.bocEur = bocEur; // 1 Eur = bocEur bocKey
  this.bocTyp = bocTyp; // noon, close rate or default when not updated
  this.bocDte = bocDte; // Get date
}


////////////////////////////////////////////////////////////////////////////
   function BoCPut(bocKey, // Currency code (text)
                   bocNam, // Currency name
                   bocRss, // Absolut Rss url at Bank of Canada
                   bocPhp, // Php at the same host
                   bocCaD, // 1 CaD = bocCaD bocKey
                   bocEur, // 1 Eur = bocEur bocKey
                   bocTyp, // noon, close rate or default when not updated
                   bocDte) // Get date

// PURPOSE: Create a currency exchange database object.
////////////////////////////////////////////////////////////////////////////
{
  var bocPos = BoCArr.length; // Position for this new entry
  BoCArr[bocPos] = bocKey;
  BoCLst[bocKey] = new BoCObj(bocKey, bocNam, bocRss, bocPhp,
                              bocCaD, bocEur, bocTyp, bocDte);
}


////////////////////////////////////////////////////////////////////////////
// BANK OF CANADA FUNCTIONS
////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////
   function BoCGetItem(inpHtm)
////////////////////////////////////////////////////////////////////////////
{ return(TxtBetween2Tag(inpHtm, "<item", "</item>")); }


////////////////////////////////////////////////////////////////////////////
   function BoCGetItemTitle(inpHtm)
////////////////////////////////////////////////////////////////////////////
{ return(TxtBetween2Tag(BoCGetItem(inpHtm), "<title>", "</title>")); }

/*----------------------------------------------------------------------------
<item rdf:about="http://www.bankofcanada.ca/rss/fx/noon/iexe0101.xml">
  <title>CA: 0.9947 USD = 1 CAD 2008-07-18 Bank of Canada noon rate</title>
  ...
  <dc:date>2008-07-18</dc:date>
  <cb:baseCurrency>CAD</cb:baseCurrency>
  <cb:targetCurrency>USD</cb:targetCurrency>
```

```
    <cb:value frequency="business" decimals="4">0.9947</cb:value>
    <cb:rateType>noon</cb:rateType>
    ...
  </item>
  ----------------------------------------------------------------------------*/

  ////////////////////////////////////////////////////////////////////////////
     function BoCGetItemDate(itm)
  ////////////////////////////////////////////////////////////////////////////
  { return(TxtReplace(XmlGetValue(itm, "dc:date"), "-", "/")); }


  ////////////////////////////////////////////////////////////////////////////
     function BoCGetItemType(itm)
  ////////////////////////////////////////////////////////////////////////////
  { return(XmlGetValue(itm, "cb:rateType")); }


  ////////////////////////////////////////////////////////////////////////////
     function RssBoCGetItemBase(itm)
  ////////////////////////////////////////////////////////////////////////////
  { return(XmlGetValue(itm, "cb:baseCurrency")); }


  ////////////////////////////////////////////////////////////////////////////
     function BoCGetItemTarget(itm)
  ////////////////////////////////////////////////////////////////////////////
  { return(XmlGetValue(itm, "cb:targetCurrency")); }


  ////////////////////////////////////////////////////////////////////////////
     function BoCGetItemValue(itm)
  ////////////////////////////////////////////////////////////////////////////
  { return(1 * XmlGetValue(itm, "cb:value")); } // Returns as numeric


  ////////////////////////////////////////////////////////////////////////////
     function BoCUpdEurCaD(bocPos,    // Must be 1 for EUR
                           cadEurXml) // Xml text

  // PURPOSE: Update the 2 main BoC objects: BoCLst["CAD"] and BoCLst["EUR"].
  //          Canadian Dollar and Euro:
  //          - must have the position 0 and 1 in the array PdbArr,
  //          - are calculate at the same time (with euros rss xml),
  //          - all other currencies depend on CaD and Eur.
  //          Side effects when finish:
  //          - calls to update the next currency (2) and
  //          - if BoCDivSlider exist insert the text EUR, CAD
  ////////////////////////////////////////////////////////////////////////////
  {
    var objCaD = BoCLst["CAD"];
    var objEur = BoCLst["EUR"];

    var cadEurItm = BoCGetItem(cadEurXml);       // Get the item information
    var cadEurVal = BoCGetItemValue(cadEurItm);  // Get the exchange value
    var cadEurDte = BoCGetItemDate(cadEurItm);   // Get the date value
    var cadEurTyp = BoCGetItemType(cadEurItm);   // Get the rate type
    var cadEurTrg = BoCGetItemTarget(cadEurItm); // Must be EUR

    if(cadEurTrg != objEur.bocKey) // Other currency?
```

```
421     { BoCAddErr("["+bocPos+"|"+cadEurTrg+"!="+objEur.bocKey+"|?];"); }
422     else
423     {
424       objCaD.bocEur = DatRound(1/cadEurVal, BoCDecPos); // 1 / Rss value
425       objEur.bocCaD = DatRound(cadEurVal,   BoCDecPos); // Rss value
426       objCaD.bocDte = cadEurDte;      // Rss date
427       objEur.bocDte = cadEurDte;      // Rss date
428       objCaD.bocTyp = cadEurTyp;      // Rss type
429       objEur.bocTyp = cadEurTyp;      // Rss type
430     }
431     BoCAddTrc("CAD EUR");
432     BoCUpdNxt(bocPos+1); // Update next currency
433   }
434
435
436   ////////////////////////////////////////////////////////////////////////////
437      function BoCUpdNxt(bocPos) // 2, 3, 4,...
438
439   // PURPOSE: Calls the next update, if there isn't call de final function.
440   ////////////////////////////////////////////////////////////////////////////
441   {
442     if(bocPos < BoCArr.length) // If there are next update it
443     {
444       var bocKey = BoCArr[bocPos];
445       var bocObj = BoCLst[bocKey];
446       BoCRssGet(bocPos, bocObj.bocRss, bocObj.bocPhp); // Ask for next Xml
447     }
448     else { BoCUpdEnd(); } // The asincronous cicle was end
449   }
450
451   ////////////////////////////////////////////////////////////////////////////
452      function BoCUpdKey(bocPos,     // Must be 1 for EUR
453                         cadKeyXml) // Xml text
454
455   // PURPOSE: Update one BoC object: BoCLst[BoCArr[bocPos]].
456   ////////////////////////////////////////////////////////////////////////////
457   {
458     var bocKey = BoCArr[bocPos];
459     var objKey = BoCLst[bocKey];
460     var eurCaD = BoCLst["CAD"].bocEur; // Change 1 Eur = n CaD
461
462     var cadKeyItm = BoCGetItem(cadKeyXml);        // Get the item information
463     var cadKeyVal = BoCGetItemValue(cadKeyItm);   // Get the exchange value
464     var cadKeyDte = BoCGetItemDate(cadKeyItm);    // Get the date value
465     var cadKeyTyp = BoCGetItemType(cadKeyItm);    // Get the rate type
466     var cadKeyTrg = BoCGetItemTarget(cadKeyItm); // Must be equal to Key
467
468     if(cadKeyTrg != objKey.bocKey) // Other currency?
469     { BoCAddErr("["+bocPos+"|"+cadKeyTrg+"!="+objKey.bocKey+"|?];"); }
470     else
471     {
472       objKey.bocCaD = DatRound(cadKeyVal, BoCDecPos);  // 1 CaD = cadKeyVal Key
473       objKey.bocEur = DatRound(eurCaD * cadKeyVal, BoCDecPos); // 1 Eur = x Key
474       objKey.bocDte = cadKeyDte; // Rss date
475       objKey.bocTyp = cadKeyTyp; // Rss type
476     }
477     BoCAddTrc(objKey.bocKey);
478     BoCUpdNxt(bocPos+1); // Update next currency
479   }
480
```

```
481
482 ////////////////////////////////////////////////////////////////////////////
483    function BoCUpdEnd() // 2, 3, 4,...
484
485 // PURPOSE: Final calls when all update cicle was end.
486 ////////////////////////////////////////////////////////////////////////////
487 {
488   HtmInjectCode("BoCTrcDiv", BoCTrcLog+".");
489   HtmInjectCode("BoCObjDiv", BoCWriteObjLst(4));
490   HtmInjectCode("BoCPriDiv",
491     BoCPriEurKey("USD", 825, 2)+"<br>"+
492     BoCPriEurKey("JPY", 825, 2));
493   HtmInjectCode("BoCTabDiv", BoCWriteFixedTable());
494 }
495
496
497 ////////////////////////////////////////////////////////////////////////////
498 // BANK OF CANADA VIEW FUNCTIONS
499 ////////////////////////////////////////////////////////////////////////////
500
501 ////////////////////////////////////////////////////////////////////////////
502    function BoCWritePhpUrl()
503
504 // PURPOSE: Writes all BoC urls for the php program (paste and copy).
505 ////////////////////////////////////////////////////////////////////////////
506 {
507   for(var arrPos=0; arrPos<BoCArr.length; arrPos++)
508   {
509     var bocKey = BoCArr[arrPos];
510     var bocObj = BoCLst[bocKey];
511     if(bocObj.bocKey != "CAD")
512     {
513       document.write('"'+bocObj.bocKey+'"=>"'+bocObj.bocRss+'"');
514       if(arrPos+1 == BoCArr.length) { document.write("<br>"); }  // The last
515       else                          { document.write(",<br>"); } // Inside
516     }
517   }
518 }
519
520
521 ////////////////////////////////////////////////////////////////////////////
522    function BoCWriteObjLst(BoCDecPos) // Precision
523
524 // PURPOSE: Writes all assigments for BoC default values with real values,
525 //          for update this program source code (paste and copy).
526 ////////////////////////////////////////////////////////////////////////////
527 {
528   var linArr = new Array();
529   for(var arrPos=0; arrPos<BoCArr.length; arrPos++)
530   {
531     var bocKey = BoCArr[arrPos];
532     var bocObj = BoCLst[bocKey];
533     var lftTxt = "BoCLst['"+bocKey+"']."
534     linArr[(3*arrPos)+0] = lftTxt+"bocCaD = " +bocObj.bocCaD+";";
535     linArr[(3*arrPos)+1] = lftTxt+"bocEur = " +bocObj.bocEur+";";
536     linArr[(3*arrPos)+2] = lftTxt+"bocDte = '"+bocObj.bocDte+"';";
537   }
538   return(linArr.join("<br>"))
539 }
540
```

```
541
542  ////////////////////////////////////////////////////////////////////////////
543     function BoCWriteFixedTable()
544
545  // PURPOSE: Returns a table with a specific order for some currencies with
546  //          4 decimal digits and with , instead the decimal point.
547  ////////////////////////////////////////////////////////////////////////////
548  {
549    var tabHtm =
550    "<table cellpadding=0 cellspacing=1 bgcolor='#0000ff' border=0>"+
551    "<tr><td>"+DatRound(BoCLst["USD"].bocEur,4)+"</td><td>USD</td></tr>"+
552    "<tr><td>"+DatRound(BoCLst["JPY"].bocEur,4)+"</td><td>JPY</td></tr>"+
553    "<tr><td>"+DatRound(BoCLst["CAD"].bocEur,4)+"</td><td>CAD</td></tr>"+
554    "<tr><td>"+DatRound(BoCLst["CHF"].bocEur,4)+"</td><td>CHF</td></tr>"+
555    "<tr><td>"+DatRound(BoCLst["GBP"].bocEur,4)+"</td><td>GBP</td></tr>"+
556    "<tr><td>"+DatRound(BoCLst["CZK"].bocEur,4)+"</td><td>CZK</td></tr>"+
557    "</table>";
558    return(TxtReplace(tabHtm,".",","));
559  }
560
561
562  ////////////////////////////////////////////////////////////////////////////
563     function BoCPriEurKey(bocKey, // Currency key
564                           priEur, // Price in Eur
565                           BoCDecPos) // Precision
566
567  // PURPOSE: Returns a text with a price change (use € to be sort).
568  ////////////////////////////////////////////////////////////////////////////
569  {
570    var objKey = BoCLst[bocKey];
571    var eurKey = objKey.bocEur;
572    var priKey = DatRound(priEur * eurKey, BoCDecPos);
573    var priTxt = priEur+" € = "+priKey+" "+bocKey;              // Text price
574    var excTxt = "€ = "+DatRound(eurKey,BoCDecPos)+" "+bocKey; // Text exhange
575    var dteTyp = objKey.bocDte+" "+objKey.bocTyp+" rate";      // Text date
576    var allTxt = priTxt+" ("+excTxt+", "+dteTyp+")";
577
578    return(allTxt);
579  }
580
581
582  ////////////////////////////////////////////////////////////////////////////
583     function BoCSelectWrite(frmNam, // Form name
584                             selKey) // Selected currency key
585
586  // PURPOSE: Write select option for all currencies in the database
587  ////////////////////////////////////////////////////////////////////////////
588  {
589    document.write(
590      "<select class='bluCon' name='bocKey' size=1 "+
591      "onChange='BoCSelectChange(document."+frmNam+");'>");
592
593    var selTxt = " ";
594    for(var arrPos=0; arrPos<BoCArr.length; arrPos++)
595    {
596      var bocKey = BoCArr[arrPos];
597      var bocObj = BoCLst[bocKey];
598
599      var picNam = bocObj.bocNam;
600      if(bocKey!=selKey) { selTxt = " "; }
```

```
601       else                { selTxt = " selected "; }
602       var optTxt = "<option"+selTxt+"value='"+bocKey+"'>"+
603                    bocKey+" | "+bocObj.bocNam+"</option>";
604       document.write(optTxt);
605     }
606     document.write("</select>");
607   }
608
609
610   ////////////////////////////////////////////////////////////////////////////
611      function BoCSelectChange(frmObj) // Form object
612
613   // PURPOSE: Update a form with currencies exchange information.
614   ////////////////////////////////////////////////////////////////////////////
615   {
616     var bocKey = frmObj.bocKey.value;
617     var bocObj = BoCLst[bocKey];
618     var priEur = frmObj.bocPri.value;
619     var bocStr = DatRound(priEur,2)+" €";
620     if(bocKey != "EUR") { bocStr =  BoCPriEurKey(bocKey, priEur, 2); }
621     frmObj.bocExc.value = bocStr;
622   }
623
624
625   ////////////////////////////////////////////////////////////////////////////
626      function BoCWritePlain(BoCDecPos) // Precision
627
628   // PURPOSE: Writes all BoC changes.
629   ////////////////////////////////////////////////////////////////////////////
630   {
631     document.write("<code>");
632     for(var arrPos=0; arrPos<BoCArr.length; arrPos++)
633     {
634       var bocKey = BoCArr[arrPos];
635       var bocObj = BoCLst[bocKey];
636       var eurKey = " 1 EUR = "+bocObj.bocEur+" "+bocKey;
637       var cadKey = " 1 CAD = "+bocObj.bocCaD+" "+bocKey;
638       var dteTyp = bocObj.bocDte+" "+bocObj.bocTyp+" rate";
639
640       document.write(eurKey+" | "+cadKey+" | "+dteTyp+"<br>");
641     }
642     document.write("</code>");
643   }
644
645
646   ////////////////////////////////////////////////////////////////////////////
647   // DATABASE
648   // IExplorer can access to any host in internet.
649   // FireFox only can access when the page and the rss are in the same host.
650   // For Php don't write the host (the same host for page and Php)
651   // /rssget.php?id=USD&ct=5 from root and works for all my differents domains
652   // over the same host.
653   ////////////////////////////////////////////////////////////////////////////
654
655   BoCPut("CAD", "Canadian Dollar", // Must be the position 0 in array BoCArr
656          "",             // Rss not needed
657          "",             // Php not needed
658          1.0,            // 1 CAD = 1 CAD
659          1.5939,         // 1 EUR = 1.5939 CAD (default value)
660          "default",      // default rate, noon rate, close rate, etc.
```

```
661         "2008/07/18"); // Get date (default value)
662
663   BoCPut("EUR", "European Euro", // Must be the position 1 in array BoCArr
664         "http://www.bankofcanada.ca/rss/fx/noon/eurocae01.xml",
665         "/rssget.php?id=EUR&ct=5",
666         0.6274,         // 1 CAD = 0.6274 EUR (default value)
667         1.0,            // 1 EUR = 1 EUR
668         "default",      // default rate, noon rate, close rate, etc.
669         "2008/07/18"); // Get date
670
671   BoCPut("USD", "U.S. Dollar",
672         "http://www.bankofcanada.ca/rss/fx/noon/iexe0101.xml",
673         "/rssget.php?id=USD&ct=5",
674         0.9947,         // 1 CAD = 0.9947 USD (default value)
675         1.5855,         // 1 EUR = 1.5855 USD (default value)
676         "default",      // default rate, noon rate, close rate, etc.
677         "2008/07/18"); // Get date
678
679   BoCPut("GBP", "U.K. pound sterling",
680         "http://www.bankofcanada.ca/rss/fx/noon/iexe1201.xml",
681         "/rssget.php?id=GBP&ct=5",
682         0.4978,         // 1 CAD = 0.4978 GBP (default value)
683         0.7934,         // 1 EUR = 0.7934 GBP (default value)
684         "default",      // default rate, noon rate, close rate, etc.
685         "2008/07/18"); // Get date
686
687   BoCPut("JPY", "Japanese yen",
688         "http://www.bankofcanada.ca/rss/fx/noon/iexe0701.xml",
689         "/rssget.php?id=JPY&ct=5",
690         106.5190,       // 1 CAD = 106.5190 JPY (default value)
691         169.7806,       // 1 EUR = 169.7806 JPY (default value)
692         "default",      // default rate, noon rate, close rate, etc.
693         "2008/07/21"); // Get date
694
695   BoCPut("CHF", "Swiss franc",
696         "http://www.bankofcanada.ca/rss/fx/noon/iexe1101.xml",
697         "/rssget.php?id=CHF&ct=5",
698         1.0185,         // 1 CAD = 1.0185 CHF (default value)
699         1.6234,         // 1 EUR = 1.6234 CHF (default value)
700         "default",      // default rate, noon rate, close rate, etc.
701         "2008/07/21"); // Get date
702
703   BoCPut("CZK", "Czech Republic koruna",
704         "http://www.bankofcanada.ca/rss/fx/noon/iexe2301.xml",
705         "/rssget.php?id=CZK&ct=5",
706         14.4634,        // 1 CAD = 14.4634 CZK (default value)
707         23.0532,        // 1 EUR = 23.0532 CZK (default value)
708         "default",      // default rate, noon rate, close rate, etc.
709         "2008/07/21"); // Get date
710
711
712   ////////////////////////////////////////////////////////////////////////////
713   // UPDATE STATIC (for easy update of default values copy from test program)
714   ////////////////////////////////////////////////////////////////////////////
715   BoCLst['CAD'].bocCaD = 1;
716   BoCLst['CAD'].bocEur = 1.5944;
717   BoCLst['CAD'].bocDte = '2008/07/22';
718   BoCLst['EUR'].bocCaD = 0.6272;
719   BoCLst['EUR'].bocEur = 1;
720   BoCLst['EUR'].bocDte = '2008/07/22';
```

```
BoCLst['USD'].bocCaD = 0.9916;
BoCLst['USD'].bocEur = 1.581;
BoCLst['USD'].bocDte = '2008/07/22';
BoCLst['GBP'].bocCaD = 0.4973;
BoCLst['GBP'].bocEur = 0.7929;
BoCLst['GBP'].bocDte = '2008/07/22';
BoCLst['JPY'].bocCaD = 106.2022;
BoCLst['JPY'].bocEur = 169.3288;
BoCLst['JPY'].bocDte = '2008/07/22';
BoCLst['CHF'].bocCaD = 1.0197;
BoCLst['CHF'].bocEur = 1.6258;
BoCLst['CHF'].bocDte = '2008/07/22';
BoCLst['CZK'].bocCaD = 14.68;
BoCLst['CZK'].bocEur = 23.4058;
BoCLst['CZK'].bocDte = '2008/07/22';
</script>


<!////////////////////////////////////////////////////////////////////////////
// Cascade Style Sheet
////////////////////////////////////////////////////////////////////////////>
<style type="text/css">
body
{
  margin:              2px 2px 2px 2px;
  font-size:           9pt;
  line-height:         1.2em;
  font-family:         lucida console;
  font-weight:         normal;
  text-align:          left;
  color:               #00ff00;
  background-color:    #000000;
}

h1
{
  margin:              0px 0px 0px 0px;
  font-size:           9pt;
  font-weight:         normal;
  color:               #0000ff;
  border-top:          1px solid #0000ff;
}

td
{
  padding:             0px 0px 0px 0px;
  background-color:    #000000;
  font-size:           9pt;
  text-align:          right;
}

a
{
  color:               #cccccc;
}

b
{
  color:               #ff0000;
  font-weight:         normal;
```

```
781   }
782
783   input
784   {
785     padding:              1px 1px 1px 1px;
786     font-size:            9pt;
787     border:               1px solid #0000ff; /*  solid groove ridge inset outset
      #e0e0e0; */
788     color:                #00ff00;
789     background-color:     #000000;
790     width:                150;
791   }
792
793   select.bluCon
794   {
795     padding:              0px 0px 0px 0px;
796     font-size:            9pt;
797     border:               1px solid #0000ff; /*  solid groove ridge inset outset
      #e0e0e0; */
798     color:                #00ff00;
799     background-color:     #000000;
800     width:                130;
801   }
802
803   #CodDiv
804   {
805     margin:               0px 0px 0px 0px;
806     width:                45%;
807     height:               100%;
808     float:                left;
809   }
810
811   #RssDiv
812   {
813     margin:               0px 0px 0px 0px;
814     width:                54%;
815     height:               100%;
816     float:                right;
817   }
818
819   </style>
820
821   <title>Gets Rss Xml exchange files (feeds) from the Bank of Canada</title>
822   </head>
823   <body>
824   <!////////////////////////////////////////////////////////////////////////////
825   // Web page body with 2 columns:
826   // - left column for code and trace (CodDiv) and
827   // - right column for Rss Xml (RssDiv)
828   ////////////////////////////////////////////////////////////////////////////>
829
830   <div id="CodDiv">
831     <h1>BANK OF CANADA (foreign exchange market for currencies)</h1>
832     <h1>Urls for each RSS (autogenerate code for the Php, all-CAD)</h1>
833     <script>BoCWritePhpUrl();</script>
834
835     <h1>Prices change examples before the conection</h1>
836     <script>
837       document.write(BoCPriEurKey("USD", 825, 2)+"<br>"+ // 825 € to $
838                      BoCPriEurKey("JPY", 825, 2));       // 825 € to Yens
```

```
    </script>

    <h1>Getting Rss Xml for each currency:</h1>
    <div id="BoCTrcDiv"> </div>

    <h1>Update code for each currency (autogenerate for javascript)</h1>
    <div id="BoCObjDiv"></div>

    <h1>The same prices change examples after the conection</h1>
    <div id="BoCPriDiv"></div>

    <h1>Select exchange example for euros to other currencies</h1>
    <form name="BoCFrm">
    <input type='text' name='bocPri' maxlength=60 value="950.25" width="530px">
    <script>BoCSelectWrite("BoCFrm","EUR");</script>
    <input type='text' name='bocExc' maxlength=60>
    </form>
  </div>

  <div id="RssDiv">
    <h1>RSS XML</h1>
    <div id="BoCXmlDiv"></div>
    <h1>Table EUR/CUR</h1>
    <div id="BoCTabDiv"></div>
  </div>




  <script>
  ////////////////////////////////////////////////////////////////////////////
  // UPDATE AJAX (begins calling Rss Xml for EUR)
  ////////////////////////////////////////////////////////////////////////////
  BoCRssGet(1, BoCLst["EUR"].bocRss, BoCLst["EUR"].bocPhp); // Look for 1st Rss
  </script>

  </body>
  </html>
```

```php
<?
////////////////////////////////////////////////////////////////////////////
// FILE    : rssget.php
// PURPOSE : Get, cache, and output contents of a Rss Xml files.
// Dado que por razones de seguridad no se pueden abrir desde Javascript
// ficheros Rss de diferente host al que alberga la pagina que contine a
// a Javascript se necesita un puente (este codigo) que residiendo en el
// mismo host que la pagina capture dichos ficheros.
// Estas restricciones de seguridad las tiene FireFox, otros navegadores y
// unas veces si y otras no Internet Explorer.
//
// Este codigo esta basado en el codigo rssfetch.php de:
// - Author: George at JavaScriptKit.com/ DynamicDrive.com
// - Created: Feb 1st, 2006. Updated: Feb 1st, 2006
// - http://www.javascriptkit.com/dhtmltutors/ajaxticker/ajaxticker2.shtml
//
// Forma de llamada: rssget.php?id=XXX&ct=9
// - id puede usarse de 2 formas:
//   - Como un identificador abreviado USD, EUR, GBP, JPY, CHF, CZK, ... que
//     accede a las direcciones almacenadas en la lista de direcciones, en
//     este caso a ficheros Xml de cambios de divisas del Bank of Canada.
//   - Como una Url completa del fichero Xml al que se desea acceder.
//     Esto deja una puerta abierta de uso general, por lo que se ha probado
//     que funciona y se ha dejado entre comentarios.
// - ct: cache time, por ejemplo, 5, en minutos.
//
// Tanto este programa como el dictorio cache con permisos de lectura y
// escritura me han funcionado bien poniendolos en la raiz /. Aunque en
// principio el directorio cache no comienza con / en un subdirectorio
// con este programa php y el directorio chache no me ha funcionado.
//
// Las paginas web con el codigo javascript que realizan las llamadas me han
// funcionado bien en cualquier parte siempre dentro del mismo host.
//
// NOTE: No puede haber ni un caracter antes del < ? inicial y ninguno despues
// del ? >. Si los hay da el error: la instrucción de proceso XML no se
// encuentra al comienzo de una entidad externa.
////////////////////////////////////////////////////////////////////////////
header('Content-type: text/xml');

////////////////////////////////////////////////////////////////////////////
// CONSTANTS
////////////////////////////////////////////////////////////////////////////
$rssLst=array // List of pairs (id, RssUrl)
(
"XXX"=>"http://www.google.com/index.html",
"USD"=>"http://www.bankofcanada.ca/rss/fx/noon/iexe0101.xml",
"EUR"=>"http://www.bankofcanada.ca/rss/fx/noon/eurocae01.xml",
"GBP"=>"http://www.bankofcanada.ca/rss/fx/noon/iexe1201.xml",
"JPY"=>"http://www.bankofcanada.ca/rss/fx/noon/iexe0701.xml",
"CHF"=>"http://www.bankofcanada.ca/rss/fx/noon/iexe1101.xml",
"CZK"=>"http://www.bankofcanada.ca/rss/fx/noon/iexe2301.xml"
);

$pthCache="cache"; // Path to cache directory

////////////////////////////////////////////////////////////////////////////
// The url of Rss Xml files are stored in rssLst.
// The argument id must contain one of: USD, EUR, GBP, JPY, CHF, CZK
// The Rss file will be stored inside the cache subdirectory.
```

```php
////////////////////////////////////////////////////////////////////////////////
$rssId=$_GET['id'];
$rssUrl=isset($rssLst[$rssId])? $rssLst[$rssId] : die("Error bad id.");
// $rssUrl=isset($rssLst[$rssId])? $rssLst[$rssId] : $rssId;
$pthFile=$pthCache. "/" . urlencode($rssUrl); // Name of file based on its Url

////////////////////////////////////////////////////////////////////////////////
// Get the minutes to cache the local Rss file based on ct argument
////////////////////////////////////////////////////////////////////////////////
$cacTim=(int) $_GET["ct"]; // Cast argument as integer (0 or greater)


////////////////////////////////////////////////////////////////////////////////
// FUNCTIONS
////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////
   function GetRssFile()

// PURPOSE: Gets the contents of an external Rss feed and saves its contents
//          to the cached file on the server.
////////////////////////////////////////////////////////////////////////////////
{
  global $rssUrl, $pthFile;
  $contents=file_get_contents($rssUrl); // Fetch the Rss file
  $fp=fopen($pthFile, "w");             // Open local file for write
  fwrite($fp, $contents);     // Write contents of Rss to cache file
  fclose($fp);                          // Close the local file
}


////////////////////////////////////////////////////////////////////////////////
   function OutRssFile()

// PURPOSE: Outputs the contents of a Rss file using the cached local file.
//          Checks if a cached version of the Rss file is available (cache
//          time), and if not, creates one first.
////////////////////////////////////////////////////////////////////////////////
{
  global $rssUrl, $pthFile, $cacTim;
  if (!file_exists($pthFile)) // If cache file doesn't exist
  {
    touch($pthFile);       // Create it
    chmod($pthFile, 0666); // Permissions
    GetRssFile();          // Read the Rss file and write to local
  }
  else if (((time()-filemtime($pthFile))/60)>$cacTim) // Minutes (/60)
  { GetRssFile(); }   // If local file time > cache time setting -> get again
  readfile($pthFile); // Return the contents of the cached local file
}


////////////////////////////////////////////////////////////////////////////////
// PROCESS
////////////////////////////////////////////////////////////////////////////////
OutRssFile();
?>
```