Demoprogramme der Software AG Benutzergruppe Österreich

Dienstag, 12. Oktober 2010
21:37

```
0010 Natural Code Examples der Software AG Benutzergruppe Österreich
0020 www.sag-bgoe.at
0030
0040 by
0050 peter.leibl@gmail.com
0060 walter.lassnig@stadt-salzburg.at
```

```
0010 * ----------------------------------------------------------------------
0020 * Generated from NATURAL XML TOOLKIT
0030 *
0040 *        'WETRQ_SE'
0050 *
0060 * DESCRIPTION
0070 *             XML serialize implementation for
0080 *             'WETTR_RQ' datastructure
0090 *
0100 * ----------------------------------------------------------------------
0110 *
0120 DEFINE DATA PARAMETER
0130 1 #XML_SERIALZE_OUTPUT (A) DYNAMIC
0140 PARAMETER USING WETTR_RQ
0150 LOCAL
0160 1 #CX                (I4)
0170 1 #CY                (I4)
0180 1 #CZ                (I4)
0190 END-DEFINE
0200 *
0210 * #XML_SERIALZE_OUTPUT := '<?xml version="1.0" encoding="ISO-8859-1"?>'
0220 *
0230 * DTD E:\NatWebs\weather_request.dtd
0240 COMPRESS #XML_SERIALZE_OUTPUT '<GetWeather'
0250   '>' INTO #XML_SERIALZE_OUTPUT LEAVING NO
0260 /* now the children
0270 COMPRESS #XML_SERIALZE_OUTPUT '<CityName'
0280   '>'
0290   GetWeather.CityName
0300   '</CityName>'
0310   INTO #XML_SERIALZE_OUTPUT LEAVING NO
0320 COMPRESS #XML_SERIALZE_OUTPUT '<CountryName'
0330   '>'
0340   GetWeather.CountryName
0350   '</CountryName>'
0360   INTO #XML_SERIALZE_OUTPUT LEAVING NO
0370 /*
0380 COMPRESS #XML_SERIALZE_OUTPUT '</GetWeather>' INTO #XML_SERIALZE_OUTPUT LEAVING NO
0390 END
```

```
0010 * returns last day of month
0020 *
0030 DEFINE DATA PARAMETER
0040 1 #DATE(D)
0050 LOCAL
0060 1 #MON1(A2)
0070 1 #MON2(A2)
0080 1 #DATE-A(A8)
0090 1 REDEFINE #DATE-A
0100   2 #JJ(N4)
0110   2 #MM(N2)
0120   2 #DD(N2)
0130 1 #I(I2)
0140 END-DEFINE
0150 *
0160 MOVE EDITED #DATE (EM=YYYYMMDD) TO #DATE-A
0170 MOVE 28 TO #DD
0180 MOVE EDITED #DATE (EM=MM) TO #MON1
0190 MOVE EDITED #DATE-A TO #DATE (EM=YYYYMMDD)
0200 FOR #I 1 4
0210   ADD 1 TO #DATE
0220   MOVE EDITED #DATE (EM=MM) TO #MON2
0230   IF #MON1 NE #MON2
0240     SUBTRACT 1 FROM #DATE
0250     ESCAPE ROUTINE
0260   END-IF
0270 END-FOR
0280 END
```

```
0010 * Balkengrafik
0020 *   by Peter Leibl, Software AG Benutzergruppe Österreich
0030
0040 DEFINE DATA
0050 PARAMETER
0060 1 #TITLE(A66) BY VALUE
0070 1 #VAL-IN(P9/1:*) BY VALUE
0080 1 #TXT-IN(A8/1:*) BY VALUE
0090 1 #COUNT(P3) BY VALUE
0100 1 #TYPE(A1) BY VALUE
0110 LOCAL
0120 1 #BREITE(P3)
0130 1 #ANZAHL(P3)
0140 1 #LINE(A67)
0150 1 REDEFINE #LINE
0160   2 #LAT(A1/1:67)
0170 1 REDEFINE #LINE
0180   2 #LFELD(A1/1:67)
0190 1 #TAB(A67/1:20)
0200 1 #I(P3)
0210 1 #N(P3)
0220 1 #Y(P3)
0230 1 #U(P3.2)
0240 1 #G(P3)
0250 1 #V(P3)
0260 1 #E(P6) INIT <1>
0270 1 #CFELD(A1)
0280 1 #MAX(P9)
0290 1 #MIN(P9)
0300 1 #VAL(P9/1:31)
0310 1 #RF(P9.5)
0320 1 #HEAD(A6)
0330 1 REDEFINE #HEAD
0340   2 #VERT(N4)
0350   2 #DELI(A2)
0360 1 #TEXT(A8)
0370 1 REDEFINE #TEXT
0380   2 #TX1(A1/1:8)
0390 END-DEFINE
0400 *
0410 IF NOT (#COUNT EQ 1 THRU 31)
0420   WRITE *PROGRAM 'Anzahl der Werte nicht zwischen 1 und 31' #COUNT
0430   STOP
0440 END-IF
0450 RESIZE ARRAY #VAL-IN TO (1:#COUNT)
0460 RESIZE ARRAY #TXT-IN TO (1:#COUNT)
0470 RESET #I
0480 ASSIGN #MIN = 999999999
0490 FOR #I 1 #COUNT          /* ERMITTELN MAXIMAL-WERT
0500   IF #VAL-IN(#I) GT #MAX
0510     MOVE #VAL-IN(#I) TO #MAX
0520   END-IF
0530   IF #VAL-IN(#I) LT #MIN
0540     MOVE #VAL-IN(#I) TO #MIN
0550   END-IF
0560 END-FOR
0570 DECIDE FOR FIRST CONDITION
0580   WHEN #MAX > 99999999 ASSIGN #E = 100000
```

```
0590   WHEN #MAX > 9999999  ASSIGN #E = 10000
0600   WHEN #MAX > 999999  ASSIGN #E = 1000
0610   WHEN #MAX > 99999  ASSIGN  #E = 100
0620   WHEN #MAX > 9999 ASSIGN    #E = 10
0630   WHEN NONE IGNORE
0640 END-DECIDE
0650 COMPUTE #RF = ( 66 - #COUNT ) / #COUNT
0660 COMPUTE #BREITE = #RF
0670 COMPUTE #ANZAHL = #BREITE * #COUNT + #COUNT
0680 *
0690 FOR #I 1 #COUNT    /* ERMITTELN HOEHE DER EINZELNEN WERTE
0700   COMPUTE ROUNDED #RF = 20 * (#VAL-IN(#I) / #MAX)
0710   COMPUTE ROUNDED #VAL(#I) = #RF
0720   IF #VAL(#I) = 0 AND #VAL-IN(#I) NE 0
0730     ASSIGN #VAL(#I) = 1
0740   END-IF
0750 END-FOR
0760 *
0770 RESET #LAT(1:66)
0780 ASSIGN #HEAD = '     I'
0790 FOR #I 20 1 STEP -1
0800   FOR #N 1 #COUNT
0810     COMPUTE #Y = #BREITE * #N - #BREITE + #N
0820     IF #VAL(#N) GE #I
0830       COMPUTE #U = #N / 2       COMPUTE #G = #N / 2
0840       IF #U NE #G
0850         ASSIGN #LAT(#Y) = '&'
0860       ELSE
0870         ASSIGN #LAT(#Y) = '/'
0880       END-IF
0890     ELSE
0900       ASSIGN #LAT(#Y) = ' '
0910     END-IF
0920   END-FOR
0930   RESET #CFELD #V
0940   FOR #N 1 #ANZAHL
0950     DECIDE FOR FIRST CONDITION
0960       WHEN #CFELD = ' ' AND #LFELD(#N) = ' ' IGNORE
0970       WHEN #LFELD(#N) = '&'
0980         ASSIGN #CFELD = '&'
0990         ADD 1 TO #V
1000         ESCAPE TOP
1010       WHEN #LFELD(#N) = '/'
1020         ASSIGN #CFELD = '/'
1030         ADD 1 TO #V
1040         ESCAPE TOP
1050       WHEN NONE IGNORE
1060     END-DECIDE
1070     IF #CFELD = '&' AND #V > 0
1080       ADD 1 TO #V
1090       MOVE '*' TO #LFELD(#N)
1100     END-IF
1110     IF #CFELD = '/' AND #V > 0
1120       ADD 1 TO #V
1130       MOVE '+' TO #LFELD(#N)
1140     END-IF
1150     IF #V > #BREITE
1160       RESET #V
```

```
1170      MOVE ',' TO #LFELD (#N)
1180    END-IF
1190  END-FOR
1200  MOVE ',' TO #LFELD (#N)
1210  MOVE #LINE TO #TAB(#I)
1220 END-FOR
1230 NEWPAGE
1240 WRITE NOTITLE '         ' #TITLE
1250 DECIDE FOR FIRST CONDITION
1260  WHEN *DEVICE EQ 'COLOR' OR #TYPE EQ 'G'
1270    FOR #I 20 1 STEP -1
1280      EXAMINE FULL #TAB(#I) FOR ' ' REPLACE FIRST WITH ','
1290      EXAMINE FULL #TAB(#I) FOR '+' REPLACE WITH ' '
1300      EXAMINE FULL #TAB(#I) FOR '*' REPLACE WITH ' '
1310      IF #I = 20 COMPUTE #VERT = #MAX / #E
1320      END-IF
1330      IF #I = 15 COMPUTE ROUNDED #VERT = ( #MAX * 3 / 4 ) / #E
1340      END-IF
1350      IF #I = 10 COMPUTE ROUNDED #VERT = ( #MAX / 2 ) / #E
1360      END-IF
1370      IF #I = 5 COMPUTE ROUNDED #VERT = ( #MAX / 4 ) / #E
1380      END-IF
1390      WRITE NOTITLE #VERT #DELI #TAB(#I) (DY=&REV/BLV?D,)
1400      ASSIGN #HEAD = '     I'
1410    END-FOR
1420  WHEN (*DEVICE EQ 'BATCH' OR EQ 'VIDEO') OR #TYPE EQ 'C'
1430    FOR #I 20 1 STEP -1
1440      IF #I = 20 COMPUTE #VERT = #MAX / #E
1450      END-IF
1460      IF #I = 15 COMPUTE ROUNDED #VERT = ( #MAX * 3 / 4 ) / #E
1470      END-IF
1480      IF #I = 10 COMPUTE ROUNDED #VERT = ( #MAX / 2 ) / #E
1490      END-IF
1500      IF #I = 5 COMPUTE ROUNDED #VERT = ( #MAX / 4 ) / #E
1510      END-IF
1520      WRITE NOTITLE #VERT #DELI #TAB(#I) (DY=&NEI/NED?D,)
1530      ASSIGN #HEAD = '     I'
1540    END-FOR
1550  WHEN NONE
1560    WRITE *PROGRAM 'HAS DETECTED UNKNOWN DEVICE:' *DEVICE
1570    TERMINATE 99
1580 END-DECIDE
1590 RESET #LINE #V
1600 FOR #I 1 #COUNT
1610  MOVE #TXT-IN(#I) TO #TEXT
1620  ADD 1 TO #V
1630  FOR #N 1 #BREITE
1640    IF #N LE 8
1650      MOVE #TX1(#N) TO #LFELD(#V)
1660    END-IF
1670    ADD 1 TO #V
1680  END-FOR
1690 END-FOR
1700 WRITE NOTITLE '         ' #LINE
1710 WRITE NOTITLE #E (EM=ZZZZZ9)
1720 * WRITE NOTITLE ' MIN/MAX-WERT :' #MIN (EM=ZZZ.ZZZ.ZZ9) '/'
1730 *  #MAX (EM=ZZZ.ZZZ.ZZ9) ',VERT. EINHEIT :' #E (EM=ZZZ.ZZ9)
1740 END
```

```
0010 * ----------------------------------------------------------------------
0020 * Generated from NATURAL XML TOOLKIT
0030 *
0040 *        'WETRS_PA'
0050 *
0060 * DESCRIPTION
0070 *             XML Parser implementation
0080 *             using PARSE XML statement for
0090 *             datastructure 'WETTR_RS'
0100 *
0110 * ----------------------------------------------------------------------
0120 *
0130 DEFINE DATA PARAMETER
0140 1 #XML_INPUT            (A) DYNAMIC BY VALUE
0150 PARAMETER USING WETTR_RS
0160 *
0170 LOCAL
0180 1 #XML_PATH            (A) DYNAMIC
0190 1 #XML_VALUE            (A) DYNAMIC
0200 *
0210 LOCAL
0220 1 #CX                  (I4)
0230 1 #CY                  (I4)
0240 1 #CZ                  (I4)
0250 END-DEFINE
0260 *
0270 * ------------------------------------------------- INCLUDE THE PARSER
0280 PARSE XML #XML_INPUT INTO PATH #XML_PATH VALUE #XML_VALUE
0290 *
0300 * DTD E:\NatWebs\weather_response.dtd
0310   DECIDE ON FIRST #XML_PATH
0320      VALUE 'CurrentWeather'
0330      RESET CurrentWeather
0340      VALUE 'CurrentWeather/Location'
0350      IGNORE
0360      VALUE 'CurrentWeather/Location/$'
0370      CurrentWeather.Location := #XML_VALUE
0380      VALUE 'CurrentWeather/Time'
0390      IGNORE
0400      VALUE 'CurrentWeather/Time/$'
0410      CurrentWeather.Time := #XML_VALUE
0420      VALUE 'CurrentWeather/Wind'
0430      IGNORE
0440      VALUE 'CurrentWeather/Wind/$'
0450      CurrentWeather.Wind := #XML_VALUE
0460      VALUE 'CurrentWeather/Visibility'
0470      IGNORE
0480      VALUE 'CurrentWeather/Visibility/$'
0490      CurrentWeather.Visibility := #XML_VALUE
0500      VALUE 'CurrentWeather/SkyConditions'
0510      IGNORE
0520      VALUE 'CurrentWeather/SkyConditions/$'
0530      CurrentWeather.SkyConditions := #XML_VALUE
0540      VALUE 'CurrentWeather/Temperature'
0550      IGNORE
0560      VALUE 'CurrentWeather/Temperature/$'
0570      CurrentWeather.Temperature := #XML_VALUE
0580      VALUE 'CurrentWeather/DewPoint'
```

```
0590      IGNORE
0600      VALUE 'CurrentWeather/DewPoint/$'
0610      CurrentWeather.DewPoint := #XML_VALUE
0620      VALUE 'CurrentWeather/RelativeHumidity'
0630      IGNORE
0640      VALUE 'CurrentWeather/RelativeHumidity/$'
0650      CurrentWeather.RelativeHumidity := #XML_VALUE
0660      VALUE 'CurrentWeather/Pressure'
0670      IGNORE
0680      VALUE 'CurrentWeather/Pressure/$'
0690      CurrentWeather.Pressure := #XML_VALUE
0700      VALUE 'CurrentWeather/Status'
0710      IGNORE
0720      VALUE 'CurrentWeather/Status/$'
0730      CurrentWeather.Status := #XML_VALUE
0740    NONE
0750      IGNORE
0760  END-DECIDE
0770 *
0780 END-PARSE
0790 *
0800 END
```

```
0010 * split text
0020
0030 define data local
0040 1 longline(a2000)
0050 1 redefine longline
0060   2 bytes(a1/1:2000)
0070 1 redefine longline
0080   2 shortline(a80/1:25)
0090 1 ix1(i4)
0100 1 ix2(i4)
0110 1 ix3(i4)
0120 1 lines(a80/1:50)
0130 end-define
0140
0150 format ls=100
0160
0170 move
0180   'TAMINO DEMO ZONE NOW ONLINE' -
0190   ' Software AGs brand new Tamino Demo Zone lets you test drive the full power' -
0200   ' of Tamino XML Server without installing the product locally. Tamino XML' -
0210   ' Server is Software AGs multi-award-winning, high-performance server with an' -
0220   'integrated native XML data store. It is an ideal integration platform for' -
0230   ' describing, storing, managing and integrating information from within and' -
0240   ' from outside an enterprise without compromising security, availability or' -
0250   'performance. Find out more about this great new offering at' -
0260   ' http://www.xmlstarterkit.com/news/tamino_demo_zone.htm'
0270   to longline
0280
0290 for ix1 1 25
0300   write shortline (ix1)
0310 end-for
0320
0330 move 80 to ix2
0340
0350 for ix3 2000 1 step -1            /* get string length
0360   if bytes (ix3) ne ' '
0370     escape bottom
0380   end-if
0390 end-for
0400
0410 repeat                            /* set delimiter
0420   for ix1 ix2 1 step -1
0430     if bytes(ix1) eq ' '
0440       move '#' to bytes (ix1)
0450       escape bottom
0460     end-if
0470   end-for
0480   compute ix2 = ix1 + 80
0490   if ix2 gt ix3
0500     escape bottom
0510   end-if
0520 end-repeat
0530
0540 separate longline into lines (*) with delimiter '#'
0550
0560 for ix1 1 50
0570   write lines (ix1)
0580   if lines (ix1) eq ' '
```

```
0590     escape bottom
0600   end-if
0610 end-for
0620
0630 end
0640
```

```
0010 * 10 sekunden nichts tun
0020
0030 define data local
0040 1 interval(i2) init <10>     /* 10 sekunden
0050 end-define
0060
0070 display *timx
0080 callnat 'USR2027N' interval
0090 display *timx
0100
0110 end
```

```
0010 * Muster für Aufruf eines Webservices
0020
0030 DEFINE DATA
0040 LOCAL USING WETTR_RQ
0050 LOCAL USING WETTR_RS
0060 LOCAL
0070 1 SOAPSEND_U (U) DYNAMIC
0080 1 SOAPSEND (A) DYNAMIC
0090 1 SOAPRECEIVE (A) DYNAMIC
0100 1 SOAPURL (A) DYNAMIC
0110   INIT <'http://www.webservicex.net/globalweather.asmx'>
0120 1 HTTP_STATUS_TEXT (A240)
0130 1 HTTP_ERROR_CODE (I4)
0140 1 XML_REQUEST (A) DYNAMIC
0150 1 XML_RESPONSE (A) DYNAMIC
0160 1 L(I4)
0170 1 X(I4)
0180 1 Y(I4)
0190 END-DEFINE
0200
0210 * Stadt und Land bereitstellen
0220 MOVE 'Wien' TO CITYNAME
0230 MOVE 'Austria' TO COUNTRYNAME
0240
0250 * XML Dokument erzeugen
0260 CALLNAT 'WETRQ_SE' XML_REQUEST GETWEATHER
0270
0280 * Namespace einfügen
0290 EXAMINE XML_REQUEST FOR '<GetWeather>'
0300 REPLACE '<GetWeather xmlns="http://www.webserviceX.NET">'
0310
0320 * SOAP Request erzeugen
0330 MOVE
0340   '<?xml version="1.0" encoding="utf-8"?>' -
0350   '<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ' -
0360   'xmlns:xsd="http://www.w3.org/2001/XMLSchema" ' -
0370   'xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">' -
0380   '<soap:Body>' -
0390   '*** myxml ***' -
0400   '</soap:Body>' -
0410   '</soap:Envelope>'
0420   TO SOAPSEND
0430
0440 * XML Dokument in den SOAP Umschlag einfügen
0450 EXAMINE SOAPSEND FOR '*** myxml ***' REPLACE XML_REQUEST
0460 examine soapsend for "" replace ""
0470 MOVE SOAPSEND TO SOAPSEND_U
0480 MOVE ENCODED SOAPSEND_U TO SOAPSEND IN CODEPAGE 'UTF-8'
0490
0500 * SOAP Request an den Server senden
0510 REQUEST DOCUMENT FROM SOAPURL
0520     WITH HEADER NAME 'Request-Method' VALUE 'POST'
0530     NAME 'Content-Type'   VALUE 'text/xml; charset=utf-8'
0540     NAME 'SOAPAction'     VALUE 'http://www.webserviceX.NET/GetWeather'
0550     DATA ALL SOAPSEND
0560     RETURN HEADER NAME ' ' VALUE HTTP_STATUS_TEXT
0570     PAGE SOAPRECEIVE
0580   RESPONSE HTTP_ERROR_CODE
```

```
0590
0600 * Returncode des Servers abfragen
0610 IF HTTP_ERROR_CODE NE 200
0620   PRINT 'HTTP Error:' HTTP_ERROR_CODE HTTP_STATUS_TEXT
0630   STOP
0640 END-IF
0650
0660 examine full SOAPRECEIVE for H'0D0A' delete
0670 move encoded soapreceive in codepage 'UTF-8' to SOAPSEND_U
0680 move SOAPSEND_U to SOAPRECEIVE
0690
0700 * utf-16 nach utf-8 konvertieren
0710 EXAMINE SOAPRECEIVE '&gt;' REPLACE '>'
0720 EXAMINE SOAPRECEIVE '&lt;' REPLACE '<'
0730
0740 * SOAP Umschlag entfernen
0750 EXAMINE SOAPRECEIVE '<CurrentWeather>' GIVING POSITION IN X
0760 EXAMINE SOAPRECEIVE '</CurrentWeather>' GIVING POSITION IN Y
0770 ADD 17 TO Y
0780 COMPUTE L = Y - X
0790 MOVE SUBSTRING(SOAPRECEIVE, X, L) TO XML_RESPONSE
0800 compress '<?xml version="1.0" ?>' XML_RESPONSE into XML_RESPONSE
0810
0820 * Antwort vom Server parsen
0830 CALLNAT 'WETRS_PA' XML_RESPONSE CURRENTWEATHER
0840
0850 * Ergebnis anzeigen
0860 PRINT NOTITLE 'Location:' LOCATION (CD=BL AD=I)
0870 PRINT 'Zeit:' TIME (CD=BL AD=I)
0880 PRINT 'Temperatur:' TEMPERATURE (CD=BL AD=I)
0890 PRINT 'Wind:' WIND (CD=BL AD=I)
0900 PRINT 'Luftfeuchtigkeit:' RELATIVEHUMIDITY (CD=BL AD=I)
0910 PRINT 'Luftdruck:' PRESSURE (CD=BL AD=I)
0920 END
```

```
0010 * wert abfragen
0020
0030 define data local
0040 1 myval(n5) init <100>
0050 end-define
0060
0070 input (ad=mil) 'Wert eingeben:' myval
0080
0090 decide on every value of myval
0100   value 100
0110     write 'exakt 100'
0120   value 90 : 110
0130     write 'zwischen 90 und 110'
0140   value 80,100,120
0150     write '80 oder 100 oder 120'
0160   none value
0170     write 'weder - noch'
0180 end-decide
0190
0200 end
```

```
0010 * letzter tag des monats
0020
0030 define data local
0040 1 mydate(d) init <*datx>
0050 end-define
0060
0070 print mydate (em=dd.mm.yyyy)
0080 callnat 'ultimo2' mydate
0090 print mydate (em=dd.mm.yyyy)
0100
0110 end
0120
0130
```

```
0010 * quick-and-dirty beispiel eines code-generators
0020 * es wird ein 'Hello World' Programm generiert und ausgeführt
0030
0040 set control 'Z'                    /* editor löschen
0050 define printer (1) output 'SOURCE'      /* alle folgenden write/display landen im editor
0060 format (1) ps=0
0070
0080 write (1) notitle nohdr "print notitle nohdr // 'Hello World !'"    /* programmcode erzeugen
0090 write (1) "print 'Visit www.sag-bgoe.at'"
0100 write (1) 'end'
0110
0120 close printer (1)
0130
0140 stack top command 'RUN PROGEN-1'
0150 stack top command 'SAVE PROGEN-1'
0160 stack top command 'RENUMBER'
0170
0180 end
```

```
0010 * Flexibler Zugriff auf Adabas Files
0020 * Alle Descriptoren sind beliebig kombinierbar
0030 *   by Peter Leibl, Software AG Benutzergruppe Österreich
0040
0050 define data local              /* Inputfelder zur Eingabe der Suchkriterien
0060 1 search-reg-num(a15)
0070 1 search-reg-num-hv(a15)       /* Zugriff mit von/bis
0080 1 search-personnel-id(a8)
0090 1 search-personnel-id-hv(a8)   /* Zugriff mit von/bis
0100 1 search-make(a20)
0110 1 search-color(a20)
0120 1 search-class(a1)
0130 1 vehicles view of vehicles    /* Software AG's Demofile vehicles
0140   2 reg-num (a15)
0150   2 personnel-id (a8)
0160   2 make (a20)
0170   2 model (a20)
0180   2 color (a10)
0190   2 year (n4)
0200   2 class (a1)
0210   2 date-acq (n8)
0220   2 curr-code (a3)
0230 1 hitlist(a8) init <'hitlist0'> /* Namen (mit fortlaufender Nummer) der Trefferliste(n)
0240 1 redefine hitlist
0250   2 hlname(a7)
0260   2 hlnmbr(n1)
0270 1 excel(a1) init <'N'>         /* Ergebnis als csv File zur Weiterverarbeitung mit Excel
0280 1 excelfile(a18) init <'c:\vehicles000.csv'>
0290 1 redefine excelfile
0300   2 fname(a11)
0310   2 fnum(n3)
0320 1 USR1052L                     /* Parameter für Userexit 1052
0330   2 OS-COMMAND(A253)
0340   2 OS-RESPONSE(I04)
0350 end-define
0360
0370 repeat
0380
0390   input (ad=milt ip=off) //       /* Suchkriterien eingeben
0400     'Fahrzeugsuche' //
0410     'Kennzeichen' 18t search-reg-num /
0420     'Personalnummer' 18t search-personnel-id /
0430     'Marke' 18t search-make /
0440     'Farbe' 18t search-color /
0450     'Klasse' 18t search-class //
0460     'Ausgabe mit Excel (J/N)' excel
0470     // 'ESC = Ende'
0480
0490   release sets                 /* Aufräumen
0500   reset initial hitlist
0510
0520   decide for every condition      /* Für jeden Descriptor Trefferliste aufbauen
0530     when search-reg-num ne ' '
0540       include flexfi_2 'fi1.' 'reg-num' 'search-reg-num' 'search-reg-num-hv'
0550     when search-personnel-id ne ' '
0560       include flexfi_2 'fi2.' 'personnel-id' 'search-personnel-id' 'search-personnel-id-hv'
0570     when search-make ne ' '
0580       include flexfi_1 'fi3.' 'make' 'search-make'
```

```
0590   when search-color ne ' '
0600     include flexfi_1 'fi4.' 'color' 'search-color'
0610   when search-class ne ' '
0620     include flexfi_1 'fi5.' 'class' 'search-class'
0630   when none
0640     reinput 'Bitte Suchkriterien eingeben oder ESC'
0650 end-decide
0660
0670 if excel eq 'J'                /* prepare for creating csv (excel) file
0680   add 1 to fnum
0690   define work file 1 excelfile type 'CSV'
0700 end-if
0710
0720 decide on first value of hlnmbr  /* merge results
0730   value 1
0740     mr1. find vehicles 'hitlist1'
0750       perform showdata          /* show results, optional write csv
0760     end-find
0770     include flexfi_3 'mr1.'
0780   value 2
0790     mr2. find vehicles 'hitlist1' and 'hitlist2'
0800       perform showdata
0810     end-find
0820     include flexfi_3 'mr2.'
0830   value 3
0840     mr3. find vehicles 'hitlist1' and 'hitlist2' and 'hitlist3'
0850       perform showdata
0860     end-find
0870     include flexfi_3 'mr3.'
0880   value 4
0890     mr4. find vehicles 'hitlist1' and 'hitlist2' and 'hitlist3'
0900         and 'hitlist4'
0910       perform showdata
0920     end-find
0930     include flexfi_3 'mr4.'
0940   value 5
0950     mr5. find vehicles 'hitlist1' and 'hitlist2' and 'hitlist3'
0960         and 'hitlist4' and 'hitlist5'
0970       perform showdata
0980     end-find
0990     include flexfi_3 'mr5.'
1000   none value
1010     reinput 'Sorry, keine Ergebnisse mit diesen Daten'
1020 end-decide
1030
1040 if excel eq 'J'
1050   close work file 1
1060   compress 'C:\Programme\Microsoft Office\OFFICE11\excel.exe' excelfile 'ASYNCH' into os-command
1070   callnat 'USR1052N' USR1052L
1080   if os-response ne 0
1090     print os-command os-response
1100     stop
1110   end-if
1120 end-if
1130
1140 define subroutine showdata
1150 if excel eq 'J'
1160   write work 1 vehicles
```

```
1170   else
1180     display reg-num personnel-id make (al=15) model (al=10) color year class
1190   end-if
1200   end-subroutine
1210
1220 end-repeat
1230
1240 end
```

```
0010 * verweilzeit innerhalb von programmteilen messen
0020
0030 define data local
0040 1 xml_fix (a) dynamic
0050 1 xml_line(a/1:50000) dynamic
0060 1 cpu1(i4)
0070 1 cpu2(i4)
0080 1 x(i4)
0090 end-define
0100
0110 write notitle nohdr 'Environment:' *opsys *natvers
0120 write *timx 'starting time'
0130 move '<line>here we go folks</line>' to xml_line (*)
0140
0150 go1. set time
0160 move *cpu-time to cpu1
0170 compress xml_line (*) into xml_fix leaving no          /* zeit vergeuden ;-)
0180 examine xml_fix for 'abc' giving position in x
0190 examine xml_fix for 'here we' replace 'lets'
0200 examine xml_fix for 'lets' replace 'here we'
0210 move *cpu-time to cpu2
0220 compute cpu2 = cpu2 - cpu1
0230 write 'cpu:' cpu2 (nl=6 ad=l) '(unit 10 ms)' / 'elapsed time:' *timd (go1.) (em=99:99:99'.'9)
0240 reset cpu1 cpu2
0250
0260 write *timx 'ending time'
0270
0280 end
```

```
0010 * statistik austria gemeindedaten abrufen
0020
0030 define data local
0040 1 gemeindedaten(a) dynamic
0050 1 geturl(a) dynamic init <'http://www.statistik.at/web_de/static/gemeinden_sortiert_nach_gemeindenamen_mit_
0060 1 httpcode(i4)
0070 1 recs(a/1:3000) dynamic
0080 1 i(i4)
0090 1 x(i4)
0100 end-define
0110
0120 REQUEST DOCUMENT FROM geturl
0130   RETURN PAGE gemeindedaten
0140   RESPONSE httpcode
0150
0160 if httpcode ne 200
0170   print 'HTTP Error:' httpcode
0180 end-if
0190
0200 examine gemeindedaten for h'0D0A' replace h'FF'
0210 separate gemeindedaten into recs (*) with delimiter h'FF'
0220   giving number in x
0230
0240 for i 1 x
0250   print i recs (i)
0260 end-for
0270
0280 end
```

```
0010 * Visualisierungsoptionen
0020 * Nationalratswahl 2008 grafische Darstellung des Wahlergebnisses
0030 *   by Peter Leibl, Software AG Benutzergruppe Österreich
0040
0050 define data local
0060 1 Parteien(A5/1:5)
0070 1 Ergebnis(n3.2/1:5)
0080 1 choice (A1) init <'K'>
0090 1 USR1052L                /* Parameter für Userexit 1052
0100   2 OS-COMMAND(A253)
0110   2 OS-RESPONSE(I04)
0120 end-define
0130
0140 move 'SPÖ' to parteien (1)
0150 move 'ÖVP' to parteien (2)
0160 move 'FPÖ' to parteien (3)
0170 move 'Grüne' to parteien (4)
0180 move 'BZÖ' to parteien (5)
0190
0200 move 35.17 to ergebnis (1)
0210 move 16.28 to ergebnis (2)
0220 move 20.83 to ergebnis (3)
0230 move 15.70 to ergebnis (4)
0240 move 4.72 to ergebnis (5)
0250
0260 repeat
0270
0280   input (ad=milt ip=off) 'Darstellung des Wahlergebnisses 2008:' choice //
0290     5t 'K = konventionell via Display' /
0300     5t 'B = Balkengrafik auf 3270 Basis, Batch Print kompatibel' /
0310     5t 'E = Balkengrafik auf 3270 extended Basis' /
0320     5t 'W = Anzeige im Browser via Google Charts' //
0330     'ESC = Ende'
0340
0350   decide on first value of choice
0360     value 'K'
0370       display parteien (*) ergebnis (*)
0380     value 'E'
0390       callnat 'BARCHART' 'Nationalratswahl 2008 Wahlergebnis Wien'
0400         ergebnis (*) parteien (*) 5 'G'
0410     value 'B'
0420       callnat 'BARCHART' 'Nationalratswahl 2008 Wahlergebnis Wien'
0430         ergebnis (*) parteien (*) 5 'C'
0440     value 'W'
0450       move 'c:\programme\internet explorer\iexplore.exe http://chart.apis.google.com/chart?cht=p3&chco=ff000(
0460         '&chd=t:35.2,16.3,20.8,15.7,4.7&chs=750x400&chl=SPOE|OEVP|FPOE|GRUENE|BZOE' to os-comman(
0470       callnat 'USR1052N' USR1052L
0480       input 'done'
0490     none value
0500       reinput 'Falsche Auswahl'
0510   end-decide
0520
0530 end-repeat
0540
0550 end
```

```
0010 * objekte aus dem web in eine natural umgebung übertragen
0020
0030 define data local
0040 1 blob(a) dynamic
0050 1 geturl(a100) init <'http://www.sag-bgoe.at/images/vorstand.jpg'>
0060 1 webrc(i4)
0070 end-define
0080
0090 REQUEST DOCUMENT FROM geturl       /* jpg-image auslesen
0100   RETURN PAGE blob
0110   RESPONSE webrc
0120
0130 if webrc ne 200
0140   input (ad=o) 'Fehler' webrc 'bei Zugriff aufgetreten'
0150 else
0160   input /// 'Bild wurde übertragen, Größe:' *length(blob) 'Bytes'
0170 end-if
0180
0190 end
```

```
0010 DEFINE DATA PARAMETER
0020 /*  XML: GetWeather -> NAT: GetWeather
0030 01 GetWeather /* PCDATA
0040   02 CityName (A) DYNAMIC
0050   02 CountryName (A) DYNAMIC /* PCDATA
0060 END-DEFINE
```

```
0010 DEFINE DATA PARAMETER
0020 /*  XML: CurrentWeather -> NAT: Curren  tWeather
0030  1 CURRENTWEATHER /*PCDATA
0040    2 LOCATION (A) DYNAMIC
0050    2 TIME (A) DYNAMIC /*PCDATA
0060    2 WIND (A) DYNAMIC /*PCDATA
0070    2 VISIBILITY (A) DYNAMIC /*PCDATA
0080    2 SKYCONDITIONS (A) DYNAMIC /*PCDATA
0090    2 TEMPERATURE (A) DYNAMIC /*PCDATA
0100    2 DEWPOINT (A) DYNAMIC /*PCDATA
0110    2 RELATIVEHUMIDITY (A) DYNAMIC /*PCDATA
0120    2 PRESSURE (A) DYNAMIC /*PCDATA
0130    2 STATUS (A) DYNAMIC /*PCDATA
0140 END-DEFINE
```

```
0010 DB: 003 FILE: 012  - VEHICLES                    DEFAULT SEQUENCE:
0020 TYPE: ADABAS
0030
0040 T L DB Name                        F Leng  S D Remark
0050 - - -- --------------------------------  - ----  - - -------------------------
0060   1 AA REG-NUM                       A   15  N D LOCAL FORMAT
0070 *     LOCAL FORMAT
0080 *     CAR'S REGISTRATION NUMBER
0090   1 AB CHASSIS-NUM                   B    4  F   NUMERIC
0100 *     NUMERIC
0110 *     UNIQUE MANIFACTURER NO FOR C
0120   1 AC PERSONNEL-ID                  A    8    D INTERN.
0130 *     INTERN.
0140 *     IDENTIFIER OF CAR USER/OWNER
0150 G 1 CD CAR-DETAILS                              LOCAL/GENERIC
0160       HD=CAR DETAILS
0170 *     LOCAL/GENERIC
0180 *     DESCRIPTION OF THE CAR
0190   2 AD MAKE                         A   20  N D LOCAL/GENERIC
0200 *     LOCAL/GENERIC
0210 *     NAME OF CAR MAKE/MANUFACTURE
0220   2 AE MODEL                        A   20  N   LOCAL/GENERIC
0230 *     LOCAL/GENERIC
0240 *     NAME OF CAR MODEL
0250   2 AF COLOR                        A   10  N D LOCAL/GENERIC
0260 *     LOCAL/GENERIC
0270 *     CAR MODEL'S COLOR NAME
0280   2 AF COLOUR                       A   10  N D LOCAL/GENERIC
0290 *     LOCAL/GENERIC
0300 *     CAR MODEL'S COLOR NAME
0310   1 AG YEAR                         N  4,0  N   INTERN.
0320 *     INTERN.
0330 *     YEAR OF THE CAR'S MANUFACTUR
0340   1 AH CLASS                        A    1  F D INTERN.
0350 *     INTERN.
0360 *     CODE FOR OWNERSHIP (P/C)
0370   1 AI LEASE-PUR                    A    1  F   INTERN.
0380 *     INTERN.
0390 *     CODE HOW CAR WAS AQUIRED
0400   1 AJ DATE-ACQ                     N  8,0  N   INTERN.
0410 *     INTERN.
0420 *     DATE WHEN CAR WAS ACQUIRED
0430   1 AL CURR-CODE                    A    3  N   INTERN.
0440 *     INTERN.
0450 *     CURRENCY OF MAINTENANCE COST
0460 M 1 AM MAINT-COST                   P  7,0  N   INTERN.
0470 *     INTERN.
0480 *     MAINTENANCE COST IN CURR.UNI
0490   1 AO MODEL-YEAR-MAKE              A   24    S LOCAL/GENERIC
0500 *     LOCAL/GENERIC
0510 *     CAR MAKE + MODEL YEAR
0520 ******DDM OUTPUT TERMINATED******
```

```
0010 if *number (&1&) eq 0
0020   reinput 'Sorry, keine Ergebnisse mit diesen Daten'
0030 end-if
0040
0050
```

```
* search range
  add 1 to hlnmbr
  compress &3& h'FF' into &4& leaving no
  &1& find number vehicles with &2& eq &3& thru &4&
    retain as hitlist
```

```
0010 * search exactly
0020     add 1 to hlnmbr
0030     &1& find number vehicles with &2& eq &3&
0040       retain as hitlist
0050
0060
```