

```

000001      h  dftactgrp(*no) debug(*xmlsax) option(*nodebugio)
000002
000003      Fsaxctl1    uf    e                      disk
000004      Fsaxdata    uf a e                      k disk
000005
000006      D  num_elems      c                      500
000007      D  xmldoc         s                      50    varying
000008      D  options        s                      20    inz('doc=file')
000009      D  currentdate    S                      D      inz(*sys)
000010
000011          // Communications area definition
000012      D myCommArea      ds
000013      D namecount       10i 0
000014      D elementdata     dim(num_elems)
000015      D name            64a overlay(elementData)
000016      D startcount      10i 0 overlay(elementData: *next)
000017      D endcount        10i 0 overlay(elementData: *next)
000018
000019
000020      D EventHandler    pr                      10i 0
000021      D commArea        likeds(myCommArea)
000022      D event           10i 0 value
000023      D pstring         *      value
000024      D stringlen       20i 0 value
000025      D exceptionID     10i 0 value
000026
000027      /free
000028
000029          // Read unprocessed records only (field prcflg = *blank)
000030      read saxctl1;
000031      dow not %eof (saxctl1);
000032
000033          // Delete any records in output file with same path and document name
000034          // If any are found, they are from a previous run
000035          dou not %found(saxdata);
000036              delete(e) (docpath:docname) saxdata;
000037          enddo;
000038
000039          // Build the document path into field "xmldoc"
000040          // The document folder will always be in the root directory
000041          clear xmldoc;
000042          xmldoc = '/' + %trim(docpath) + '/' + %trim(docname);
000043
000044          // If an error occurs, mark the process flag with an 'E'
000045          monitor;
000046              // Begin the parsing, processing is similar to a subroutine.
000047              // Control will be passed to the "handler" for each event encountered.
000048
000049          xml-sax %handler(EventHandler : myCommArea) %xml(%trim(xmldoc):options);
000050
000051          on-error *program;
000052              prcflg = 'E';
000053          endmon;
000054
000055          // Mark this control file record as processed.
000056          // Processed values are 'E' for Error, 'X' for no errors found.
000057          if prcflg = *blanks;
000058              prcflg = 'X';
000059          endif;
000060
000061          prcdat = currentdate;
000062          prctim = %time;
000063
000064          update saxctlr;
000065
000066          // Process the next record in saxctl1 file
000067      read saxctl1;
000068      enddo;
000069

```

```

000070      // When here at eof, end job
000071      *inlr = *on;
000072      return;
000073
000074 /end-free
000075
000076      // This handler will be called each time a new event is encountered.
000077      // A long complex document can easily call it hundreds
000078      // or thousands of times.
000079
000080 P EventHandler      b
000081 d                  pi          10i 0
000082 D commArea          likeds(myCommArea)
000083 D event              10i 0 value
000084 D pstring            * value
000085 D stringlen          20i 0 value
000086 D exceptionID        10i 0 value
000087
000088 D string              s          65535a based(pstring)
000089 D returnCode          s          10i 0 inz(*zero)
000090 D name                s          256a varying
000091 D element              s          10i 0
000092
000093
000094 /free
000095
000096      select;
000097
000098      // Communications area definition
000099      when event = *XML_START_DOCUMENT;
000100          clear commArea;
000101
000102      when event = *XML_START_ELEMENT;
000103          name = %subst(string : 1 : stringlen);
000104          clear *nokey saxdatar;
000105          datatype = 'Start_Element';
000106          data = name;
000107          write saxdatar;
000108
000109      when event = *XML_END_ELEMENT;
000110          name = %subst(string : 1 : stringlen);
000111          clear *nokey saxdatar;
000112          datatype = 'End_Element';
000113          data = name;
000114          write saxdatar;
000115
000116      when event = *XML_ATTR_NAME;
000117          name = %subst(string : 1 : stringlen);
000118          clear *nokey saxdatar;
000119          datatype = 'Attr_Name';
000120          data = name;
000121          write saxdatar;
000122
000123      when event = *XML_ATTR_CHARS;
000124          name = %subst(string : 1 : stringlen);
000125          clear *nokey saxdatar;
000126          datatype = 'Attr_Chars';
000127          data = name;
000128          write saxdatar;
000129
000130      when event = *XML_CHARS;
000131          name = %subst(string : 1 : stringlen);
000132          clear *nokey saxdatar;
000133          datatype = 'XML_Chars';
000134          data = name;
000135          if data > *blanks;
000136              write saxdatar;
000137          endif;
000138

```

```
000139         ends1;  
000140         return returnCode;  
000141  
000142     /end-free  
000143  
000144 p EventHandler     e
```