PL/B and XML

Bill Keech

Sunbelt Computer Systems

OVERVIEW

- What Is XML ?
- XML Element
- XML Attribute
- XML Document
- Listview/Treeview XML
- XFILE Support
- Future XML Support
- Questions

What Is XML ?

- Extensible Markup Language (XML) is a standard for encoding data
- XML allows data to be marked up with tags
- XML starting tag is a string of text enclosed using the < and > characters
- XML ending tag is a string of text with a leading / that is enclosed using the < and > characters

What Is XML ?

- A matching pair of XML starting and ending tags constitutes an XML element
- Starting and ending tags must match
- The tag name is also the name of the element

```
<PHONE>(111) 123-1235</PHONE>
```

XML Element

- XML elements Identify named sections of data
- May contain other XML elements, XML attributes, or text

```
<PERSON>
   <NAME>Bill</NAME>
   <PHONE>(111) 123-1235</PHONE>
</PERSON>
```

XML Element

- Relationships between elements are described in the terms of child and parent

- XML elements can contain both XML attributes and text at the same time

XML Attribute

- Allows the addition of information about an XML element
- XML attributes are in the form of name=value pairs
- Attributes are placed in the starting tag of the element after the tag name

XML Attribute

- Attributes are separated by using a leading space character
- Values are enclosed in either single or double quotes

```
<PERSON PERSONID="5622">
   <NAME>John Smith</NAME>
</PERSON>
```

XML Document

- Start with an XML declaration line

```
<?xml version="1.0"?>
```

- May have an optional XML document type declaration

```
<!DOCTYPE .... >
```

- Must have a single document or root XML element

XML Document

- Can contain XML comment lines

```
<!-- comment goes here -->
```

- Can contain XML processing instruction lines

```
<?name text?>
```

Sample XML File

```xml
<?xml version="1.0" standalone="yes"?>
<PurchaseOrder>
  <Customers>
    <CustomerID>CHOPS</CustomerID>
    <Orders>
      <OrderID>10966</OrderID>
      <OrderDetails>
        <ProductID>37</ProductID>
        <Quantity>8</Quantity>
      </OrderDetails>
      <OrderDetails>
        <ProductID>56</ProductID>
        <Quantity>12</Quantity>
```

Sample XML File

```
      </OrderDetails>
    </Orders>
    <CompanyName>Chop-suey Chinese</CompanyName>
  </Customers>
 <Shippers>
   <ShipperID>1</ShipperID>
   <CompanyName>Speedy Express</CompanyName>
  </Shippers>
  <Products>
   <ProductID>37</ProductID
   <UnitsInStock>11</UnitsInStock>
  </Products>
</PurchaseOrder>
```

Listview/Treeview XML

- New LoadXmlFile and SaveXmlFile methods
- LISTVIEW object stored or loaded as a simple table and row XML element set
- TREEVIEW object stored or loaded as a tree structure
- LISTVIEW data can be filtered on input by using $LV_XMLRD_USECOLUMNS flag
- TREEVIEW data can be optionally stored

Listview/Treeview XML

- Object information optionally stored as XML attributes

```
<Root>
 <Row lvi=0 lvp=0>
   <Order_Id lvw="80" lvf="16384"> 107</Order_Id>
   <Cust_Id lvw="117″ lvf="16384">WITAE</Cust_Id>
 </Row>
</Root>
```

XFILE Support

- XFILE is an abstract representation of an XML document

- XFILE is memory resident

- XML elements are broken into records and data fields

XFILE Support

- A record is:
  - 1. An XML element that has attributes
  - 2. An XML element that has child elements
- A record can contain other records, and data fields

XFILE Support

- A data field is:
  - 1. An XML element that has no attributes or child elements
  - 2. An XML attribute of an XML element considered to be a record
- A repeating data field is a data field array

XFILE Support

- A XML element with attributes, and text has a data field named "Record_Text" to reference the text
- Records are group into record sets
- The record name (element tag name) is the record set name

XFILE Support

- Records within records are also grouped into record sets

- A XFILE corresponds to one record set

- Multiple XFILEs might be required to access an XML document

XFILE Statements

- New XFILE data type
- OPEN statement used to load an XML document
- PREPARE statement used to create a new XML document
- FLUSH statement used to save a copy of the XML document to disk
- FPOSIT statement used to obtain a recordset position

XFILE Statements

- REPOSIT statement used to set a recordset position
- POSITEOF statement used to set the position to the end of a recordset
- READ statement used to read records from a recordset
- READKG statement used to read records from a recordset
- READKGP statement used to read records

XFILE Statements

- WRITE statement used to write records into a recordset
- INSERTXML statement used to insert records into a recordset
- UPDATE statement used to update records in a recordset
- DELETE statement used to remove records from a recordset

XFILE Statements

- GETFILE statement used to set information for a XFILE
- SETFILE statement used to set information for a XFILE
- CLOSE statement used to terminate XFILE operations

XFILE/FILE Similarities

- Positioning a XFILE is the same as a fixed length record FILE
- DELETE statement removes a record but causes no positional change
- WRITE statement will overwrite an existing record

XFILE/FILE Similarities

- READ statement list controls are *LC, *LL, *PL, and *UC
- WRITE/UPDATE statement list controls are *LC, *LL, *PL, *UC, *ZF and *ZS

XFILE/FILE Differences

- XFILEs can be initialized by OPEN, PREPARE, or a READ statement
- XFILEs are independent from each other
- No shared access between XFILEs
- No pre-allocation on writes past the end of the recordset

XFILE/FILE Differences

- Variables in an I/O list are referenced by field name, not position in list
- A record may contain another record set that must be referenced by another XFILE
- A record may be written or updated containing another record set (XFILE)

XFILE Example 1

```
<?xml version="1.0"?>
<PEOPLE>
   <PERSON PERSONID="5622">
      <NAME>John Smith</NAME>
      <ADDRESS>512 Every Road</ADDRESS>
      <PHONE>(111) 123-6645</PHONE>
   </PERSON>
   <PERSON PERSONID="6109">
      <NAME>James Jones</NAME>
      <ADDRESS>513 Every Road</ADDRESS>
      <PHONE>(111) 123-4567</PHONE>
   </PERSON>
</PEOPLE>
```

XFILE Example 1

```
People    XFILE
Person    XFILE
  OPEN    People,"people.xml"
  READ    People,SEQ;PERSON=Person
  READ    Person,SEQ;PERONID=id,NAME=name
  UPDATE  Person;ADDRESS=newAddress
  CLOSE   Person
  CLOSE   People
```

XFILE Example 2

```
<?xml version="1.0"?>
<LIBRARY>
    <BOOK>
        <NAME>Programming with PL/B</NAME>
        <TYPE>Computer</TYPE>
    </BOOK>
    <MAGAZINE>
        <NAME>Visual PL/B World</NAME>
        <ISSUE>45</ISSUE>
    </MAGAZINE>
    <BOOK>
        <NAME>My Life As A PL/B Programmer</NAME>
        <TYPE>Adventure</TYPE>
    </BOOK>
</LIBRARY>
```

XFILE Example 2

```
Library     XFILE
Book        XFILE
Mag         XFILE
  OPEN      Library,"library.xml"
  READ      Library,SEQ;BOOK=Book,MAGAZINE=Mag
  WRITE     Book,"-3";NAME="PL/B Samples”:
            TYPE="Programming"
  WRITE     Mag,"-3";Name="Visual PL/B World”:
            ISSUE="46"
  UPDATE    Library;BOOK=Book,MAGAZINE=Mag
  CLOSE     Book
  CLOSE     Mag
  CLOSE     Library
```

XFILE Example 3

```
<?xml version="1.0"?>
<LIBRARY>
    <BOOK>
        <NAME>Programming with PL/B</NAME>
        <TYPE>Computer</TYPE>
    </BOOK>
    <MAGAZINE>
        <NAME>Visual PL/B World</NAME>
        <ISSUE>45</ISSUE>
    </MAGAZINE>
    <BOOK>
        <NAME>My Life As A PL/B Programmer</NAME>
        <TYPE>Adventure</TYPE>
    </BOOK>
</LIBRARY>
```

XFILE Example 3

```
Library      XFILE
Book         XFILE
Mag          XFILE
   OPEN      Library,"library.xml"
   READ      Library,SEQ;BOOK=Book,MAGAZINE=Mag
   CLOSE     Library
   WRITE     Book,"-3";NAME="PL/B Samples”:
             TYPE="Programming"
   WRITE     Mag,"-3";Name="Visual PL/B World”:
             ISSUE="46"
   PREPARE   Library,"libmod.xml"
   WRITE     Library;BOOK=Book,MAGAZINE=Mag
   CLOSE     Library
   CLOSE     Book
   CLOSE     Mag
```

Future XML Support

- XML schema integration with XFILEs
- XML schema integration with FILEs, IFILEs, and AFILEs
- PL/B Forms stored in XML
- Printer Forms in XML for PRTPAGE

QUESTIONS?

That’s All!!