



XML THUNDER

Modernize your NonStop COBOL Applications with XML Thunder

September 29, 2009

Mike Bonham, TIC Software

John Russell, Canam Software

Agenda



- XML Overview
- XML Thunder overview
- Case Studies
- Q & A

XML Standard



- XML is becoming the de-facto standard for business-to-business data exchange among different industries:
 - Financial
 - Commercial
 - Banking
 - Insurance
 - Retail

Examples of XML Standards



- Banking
 - SWIFT interface
- Financial
 - SEPA
- Retail and Commercial Services
 - XML to replace EDI
- Insurance
 - ACORD
- Financial Brokerage
 - FIX and XML
- Travel
 - OTA (OpenTravel Alliance)



A Quick Review of XML

XML Overview



- XML – Extensible Markup Language
 - A self-describing data representation ‘language’ including both data and meta-data
- What it does
 - Allows designers to create their own customized tags, enabling the definition, transmission, validation and interpretation of data between applications and organizations

Example

```
<?xml version="1.0"?>
<order
number="C10072"
orderDate="20020501" >
<orderline id="1013"
quantity="2" price="500"
/>
<totalamount> 1005.99
</totalamount>
</order>
```

Self-Describing Data



- **Sequential File Sample Data**

CA-10072 Smith & Smith 7118234 20020501 1013 2 @|| AD1D

- **EDI Sample Data**

PO1* CA-10072 * Smith & Smith *EA*500.00*BP*7118234*EC*A*VP*1013
*1005.99

- **XML Sample Data**

```
<?xml version="1.0"?>
<order number="CA-10072" custname="Smith & Smith" custno="7118234"
orderDate="20020501" >
<orderline id="1013" quantity="2" price="500" />
<totalamount> 1005.99 </totalamount>
</order>
```

XML Schemas



- Contain a much more detailed description of what comprises a valid document
- Support simple and complex data types
 - Includes built-in types
string, integer, decimal, boolean, byte, duration, dateTime etc.
 - You can declare your own new types
- Namespace aware element and attribute declarations
- Element occurrence constraints
 - minOccurs, maxOccurs

XML Schema Example



XML Document

```
<?xml version="1.0"?>
<note>
  <to>John</to>
  <from>Mary</from>
  <heading>Reminder</heading>
  <body>Don't forget me</body>
</note>
```

XML Schema

```
<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.w3schools.com"
  xmlns="http://www.w3schools.com"
  elementFormDefault="qualified">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



Why is XML important?

- Common Standard
 - Business partners exchange data in XML format
 - Becoming pervasive - practically all systems that interact with the Internet will need to deal with XML
 - Support increasingly built-in to programming languages such as Java and .Net

Common Misconceptions about XML



- “NonStop can’t handle XML. You need Java or Windows to do that”
- “I need to learn a lot about XML before I can write a COBOL program to handle XML”
- “It looks like just a simple set of tags. I can write a program to do that easily”

XML applications can be developed easily on the NonStop using COBOL using the right tool...



The solution is...

XML Thunder

What is XML Thunder?

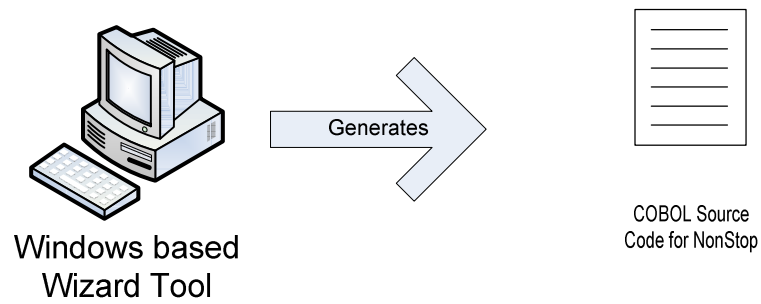
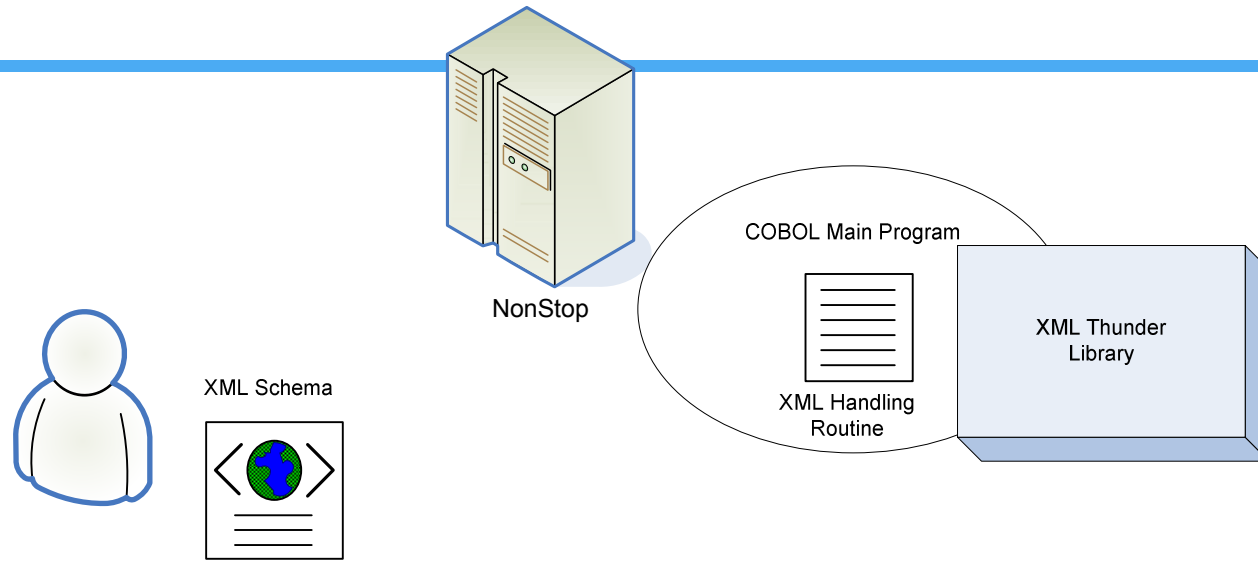


- Development tool that generates COBOL code for
 - reading / parsing XML documents
 - creating XML documents
- Windows based development tool
 - Visual designer and code generator
- Full life cycle solution
 - Use in development and maintenance

Why Use XML Thunder?



- Allows COBOL developers to work with XML quickly and easily
 - Generates code to read and write XML documents in minutes/hours instead of days/weeks/months
- Addresses the complexities of XML processing
 - Schema understanding / interpretation
 - Optional nodes vs. required nodes
 - Unbounded repeating structures
 - Nested repeating structures
- Make changes easily without major re-write of code



How does XML Thunder work?



- The Basics (toolset)

1. **Derive**

- The XML structure and Interface data structure



2. **Bind**

- Map XML nodes to language specific program data fields (IDS)



3. **Generate**

- Generate source code to consume or create XML documents
 - COBOL
 - C

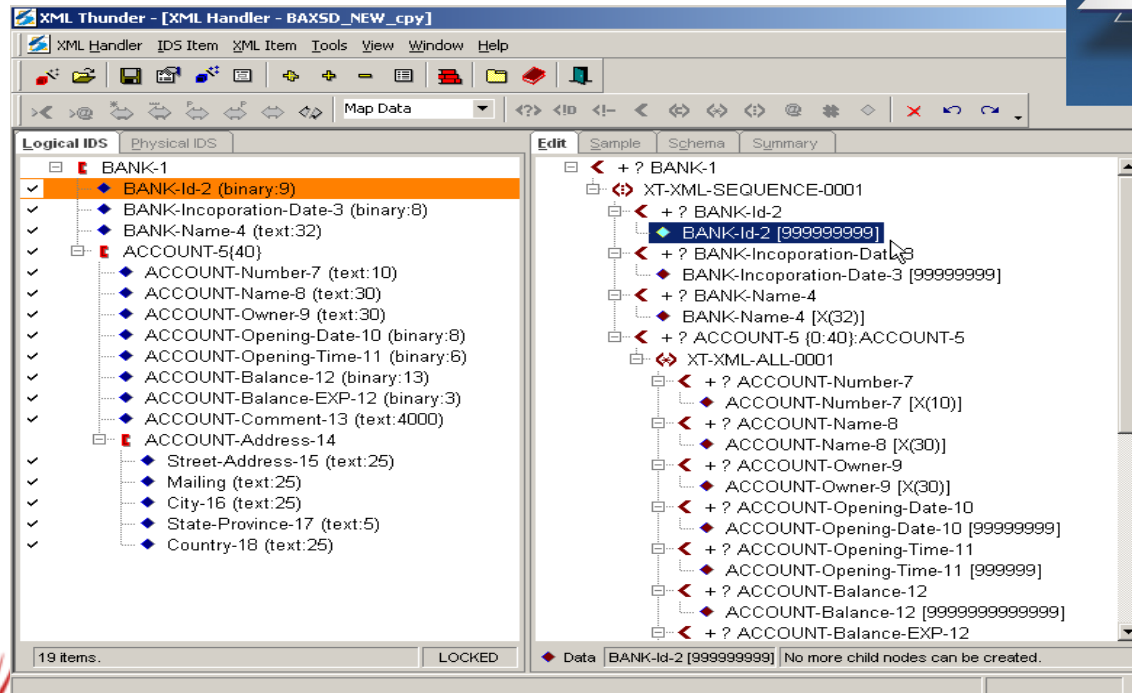


1. Derive



2. Bind

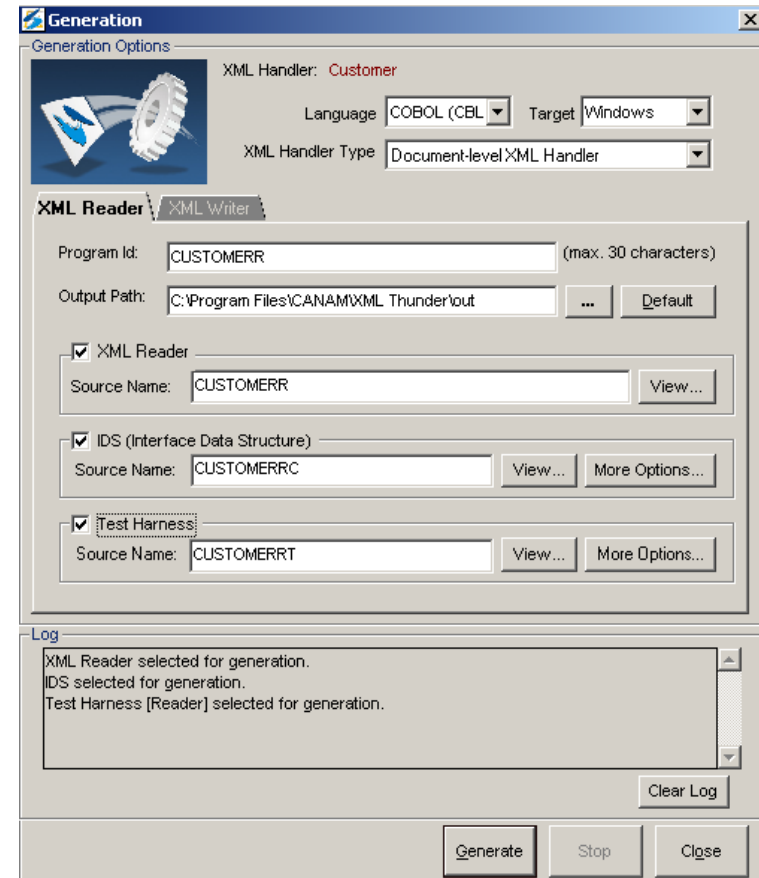
- Automatic binding (mapping)
 - After deriving one side
 - Most cases
- Manual binding (mapping)



3. Generate



- XML Handler
 - XML Reader
 - XML Writer
- IDS (optional)
- Test Harness (optional)
 - Allows testing of XML Handler
 - Model calling code (copy/paste)



Use it in your COBOL Program



Use the COPYBOOK and the Handler in your own COBOL program.

- **Populate the working storage fields in your program.**
- ```
MOVE "ABC COMPANY" to ACCOUNT-NAME....
```
- **When ready, call the Handler to handle the XML document automatically**
- ```
CALL "BAXMLW" <==== Generated Handler Routine
```
- ```
USING
```
- ```
CANAM-XML-DATA <==== Data in Working Storage
```
- ```
CANAM-XML-BUFFER <==== Output XML Buffer
```
- ```
CANAM-XML-STATUS <==== Returned Status
```
- ```
END-CALL.
```

# Case Study



- Major food service distributor
  - Long time NonStop User
  - Mostly COBOL developers
  - Uses iTP Web Server for web application
- Requirement
  - External supplier wants to use XML to send order transactions
  - Needs to get project done in 10 weeks

# Case Study



- Challenge
  - COBOL programmer not trained on XML
  - Attempted to code it manually
  - Hand coded for 6 weeks using sample XML documents with only limited success
- What we did..
  - Worked with XML schema, not sample doc
  - Found vendor's schema is not "typical"
    - Multiple XSD
    - Missing root element!
    - "Yes, we know about it, but...no, we won't change"

# Case Study



- What we did...
  - Fixed the vendor's schema
  - Used XML Thunder to generate XML handling code
  - Support the developer to get the COBOL programs tested and working completely, within 5 days
- Results
  - Successful project implementation
  - Now, the whole development team can quickly develop XML handling COBOL programs
- Customer Benefits
  - *Always work with the schema*
  - *Project completed on time*
  - *Make sure it is valid*

# Success Story



- Royal Bank of Canada
- Requirement
  - Interfacing to Swift
- Solution
  - XML Thunder
- Case Study
  - <http://nonstop.xmlthunder.com>





# XML Thunder Solution



- XML Thunder runs on the NonStop very easily and efficiently
- XML Thunder generates code for your COBOL or C program to handle XML documents. You don't have to be an XML expert
- XML Thunder provides schema validation and logic to make sure your XML documents are correct

*Perfect tool for your NonStop XML needs!*

*Get your NonStop XML application up and running in a matter of HOURS instead of DAYS!*

# Common Questions

---



- What Versions of COBOL does it support?
- How large of document can XT handle?
- What if I have bad data in the buffer?
- What kind of performance does it have?



Q & A

# Thank You!



For questions, contact  
Mike Bonham  
402-344-8888  
[Mike.Bonham@TICSoftware.com](mailto:Mike.Bonham@TICSoftware.com)

