

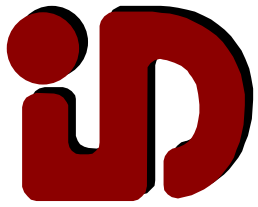


Inventive Designers n.v.



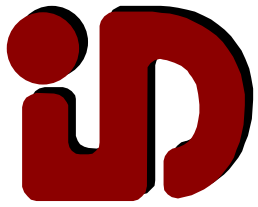
XML for RPG

How to integrate
both worlds



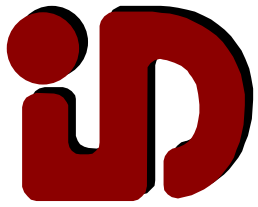
XML Basics

- eXtended Markup Language
- Tag based language
- Describes data
- Well formed XML Document
 - Begin tag must have end tag
 - Tags must be nested.
 - ▲ `<tag1 > this<tag2>is not correct</tag1 > </tag2>`



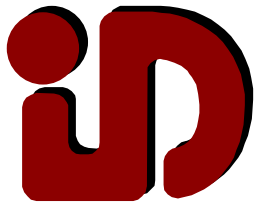
XML - Basics

- DTD defines the tags and their relationship
- Valid XML document must
 - be a well formed document
 - follow the DTD rules
- Widely adopted
 - IBM, Lotus, Microsoft, Netscape, Sun, Adobe
 - W3C's DOM APIs
- Is very useful for programs using XML documents

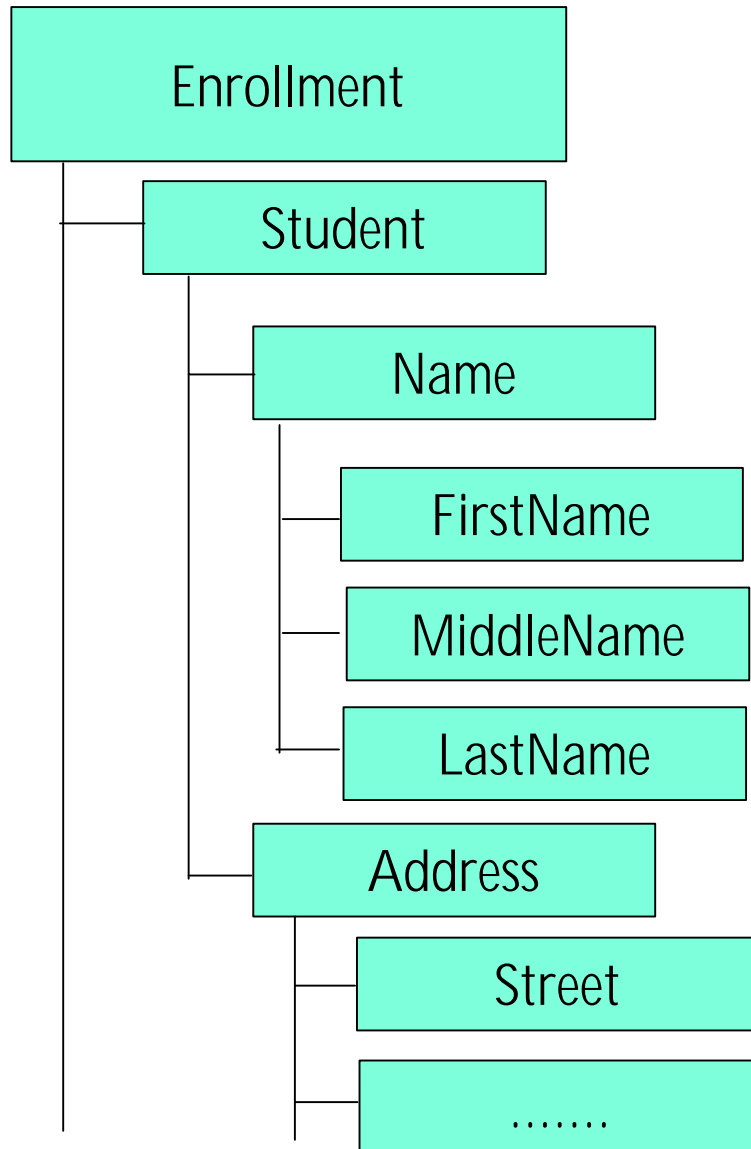


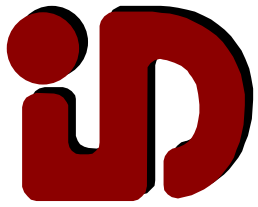
XML Parsing

- XML document is ASCII string
- Is practically not useable as such in a program
- Parsers are "tools" that
 - represent the XML document as a tree
 - provide access methods to that tree
- Used by applications
 - f.e. Internet Explorer 5.0



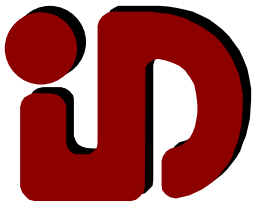
XML Parsers





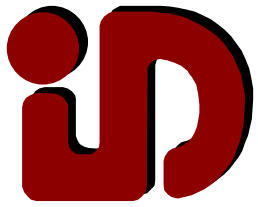
C++ example

```
QxmlIDOM_Element_getElementsByTagName  
( DOM_Element obj, void* name, int stringind, int inlen )
```



ILE Basics

- Modules contain procedures
- Modules are bound during program creation
- Procedure calls are static calls (fast)
- Service programs are bound when first called
- Procedure calls are prototyped



ILE Basics (*continued*)

■ Procedure calls return values

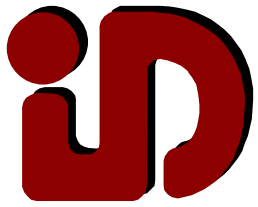
■ EVAL name@ = GetNameAddress (Student)

■ GetNameAddress is procedure returning a pointer field

■ GetNameAddress must be prototyped

▲ D PR * EXTPROC('GetNameAddress')

▲ D 256 Value



RPG for XML toolbox

- Service program XML4PR310
- Contains ILE procedures
- Each procedure maps a C++ class/method
- Procedure prototypes are provided in /copy source



RPG for XML toolbox *(continued)*

■ Examples

■ Parse XML document from XML file

- ▲ Name procedure: Qxml_DOMParser_parse_SystemID
- ▲ Parameters: pointer to parser, filename, code page
- ▲ Returns pointer to document
- ▲ Usage: EVAL domdoc@=Qxml_DOM_Parser....



RPG for XML toolbox (continued)

■ Create list of elements

▲ Name procedure:

QxmlDOM_Document_getElementetsByTagNamePrototype

▲ Parameters: pointer to document ...

▲ Returns pointer to table of element addresses

▲ Usage: EVAL elemList@=QXMLDOM_Document

■ Get name of element

▲ Name procedure: QxmlDom_Node_getNodeName

▲ Parameters: pointer to node

▲ Returns pointer to name



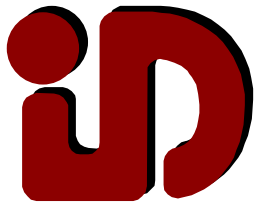
XML for RPG SRVPGM

■ Advantages

- It works

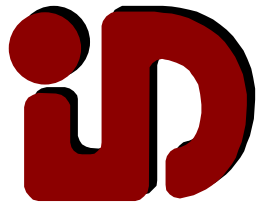
■ Disadvantages

- Usage of pointers
- Look and feel of C++ Parser
- Not complete
- Requires object management
- Not documented



XML for RPG White Paper

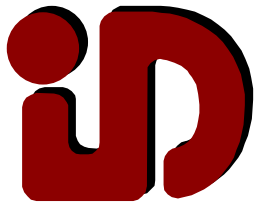
- Describes sample program
- Written by C++ programmers
- Contains some logic errors
- Not well documented



XML for RPG

Options

- Don't use the parser
 - Do it yourself
 - Not "doable"
- Use the XML for RPG wrapper provided by IBM
 - Medium ILE skills are pre-requisite
 - User Object Management!
- Use classical RPG procedures
 - Provided by Inventive Designers
 - Limited ILE skills required



XML for RPG

■ Classical RPG interface

- Service Program containing RPG procedures

- Example

 - ▲ AddElement

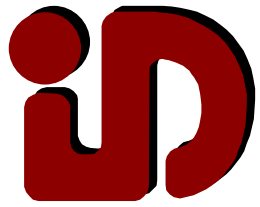
 - Parameter : Name of element, Name of last parent

 - ▲ AddElementtext

 - Parameter: Field containing text, Name of the element

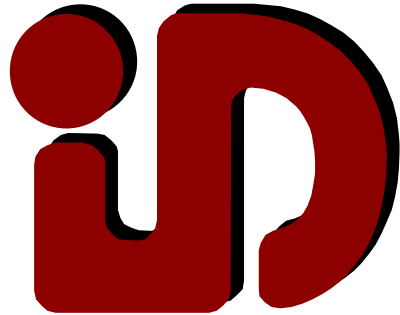
- No pointer management

- No user object management



XML for RPG Bibliography

- "Just XML" by John E. Simpson (Prentice Hall)
- <http://www.alphaworks.ibm.com/tech/xml4rpg>
- <http://www.inventivedesigners.com>
- IBM/Inventive Designers training sessions



for more information, contact

Willy Bernaerts, at
Inventive Designers n.v.

Sint-Bernardsesteenweg, 552
B-2660 Antwerp - Hoboken
Belgium

Tel: +32 3 821 01 70
Fax: +32 3 821 01 71

E-mail: Willy_Bernaerts@inventivedesigners.com

visit our website <http://www.inventivedesigners.com/>