Practical Usage of UGS PLMXML File Format.

Bronek Grzegoerk

EDS

Bronek.Grzegorek@xerox.com

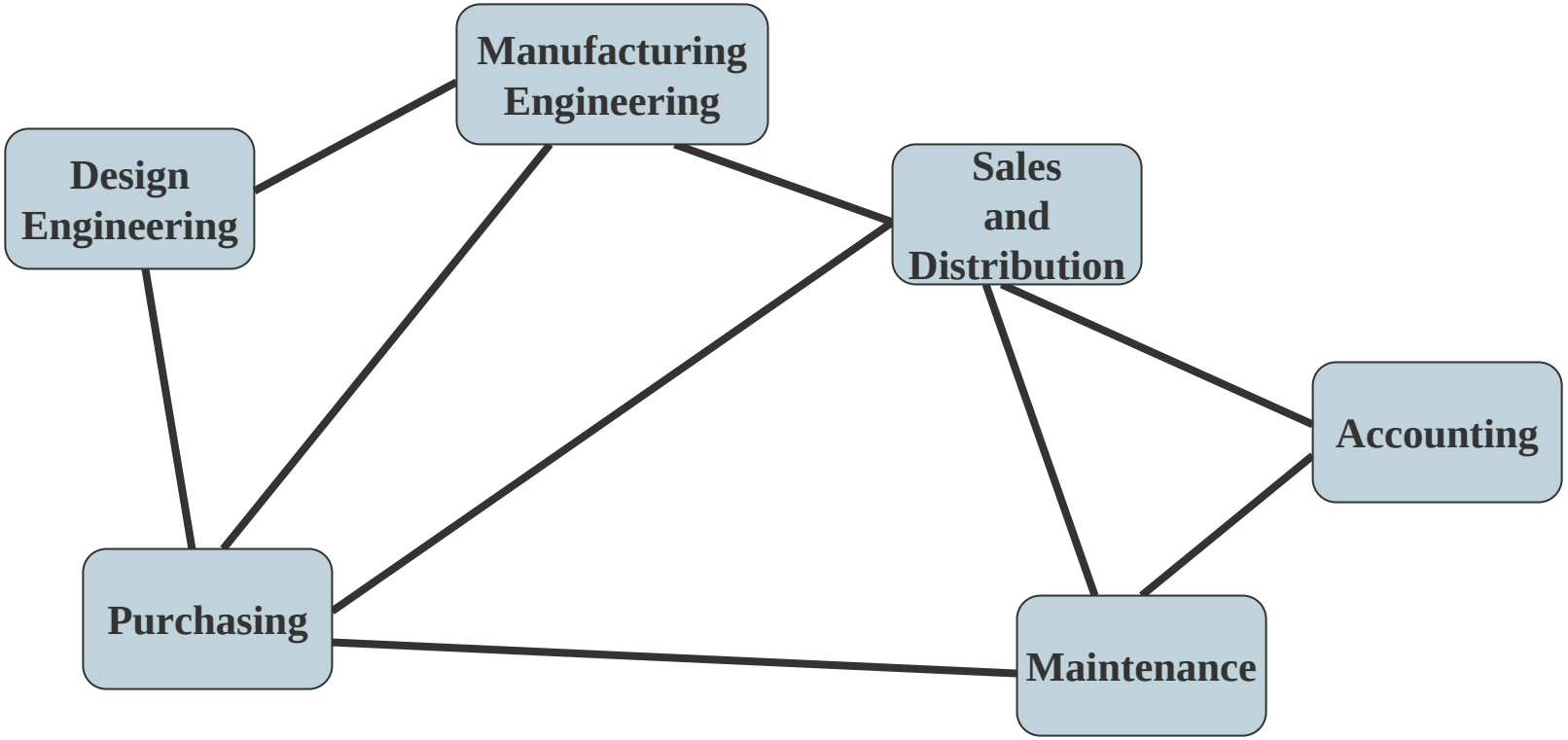
(585) 422-6129

Premium Partners:

Agenda

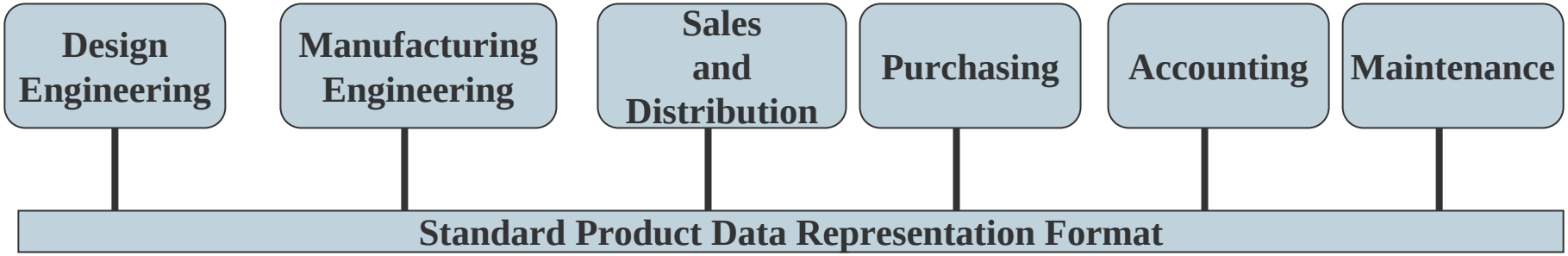
- Product Data Exchange Standards.
- XLM technology overview.
- PLMXML data schema overview.
- MSXML Parser with Internet Explorer
- How to read and manage PLMXML files.
- Demo of practical examples.

Standards



Product Data requires standard representation for interoperability.
Propitiatory formats needs to be balanced by common standard.

Common Standard



- Support of interoperability and integration.
- Enable different products and systems made to work together.
- Provide specifications explaining how information should look.

Types of Standards

Three types of standards for product information exchange:

- Open Standards.
- Industry Standards
- De Facto Standards

Open Standards

Open Standards

Developed by consensus in an industry group, for example:

STEP - developed by ISO .

XML developed by the World Wide Web Consortium (W3C)

Industry Standards

Industry Standards

Technologies that are commonly used, but are not open or democratically managed by a group of users.

The **Java™** is a well-known example of an industry standard.

De Facto Standards

De Facto Standards

They are in wide use because of their value or association with other technologies, standards organization, for example :

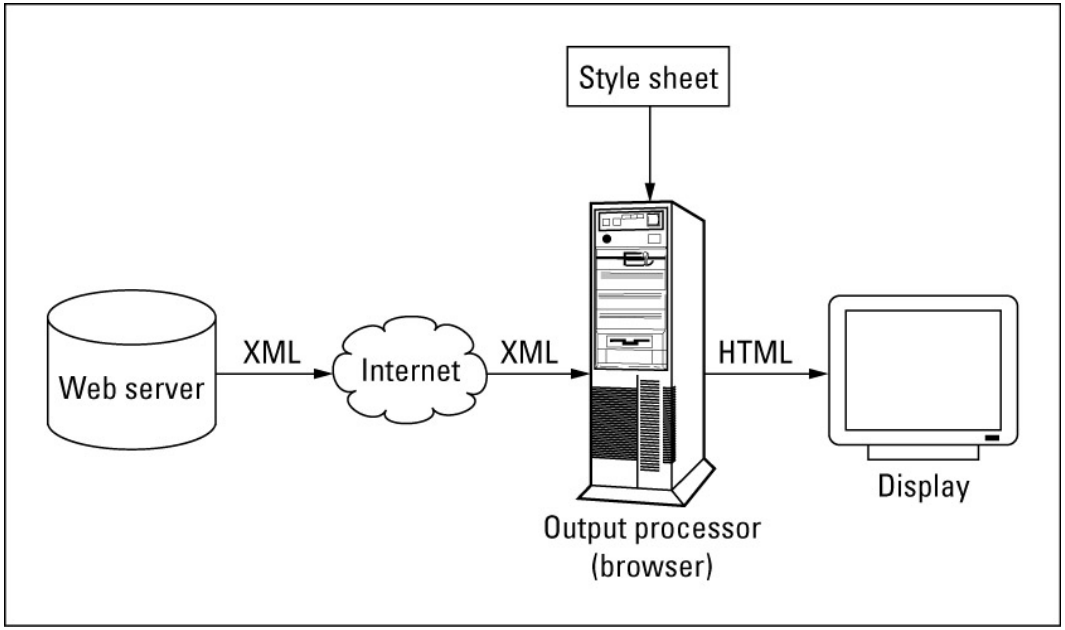
DXF - developed by AutoDesk.

PDF – Developed by Adobe Systems.

PLM XML

- Implements XML format for data exchange interoperability.
- Similar to STEP product data definition (schema).
- It is becoming industry “de facto” standard.

XML (Extensible Markup Language).



- Technology for WWW applications.
- Sponsored by World Wide Web Consortium
- Design to simplifies business-to-business transactions.

XML vs. HTML

HTML

- Lacks fine controls. It is too “forgiving”.
- Browsers’ displays vary.
- Design for data presentation (rendering)
- Not good for complex data interchange

XML vs. HTML

XML

- Applications can easy access and extract information.
- Enables robust database contents exchange
- Data can be used in HTML pages.
- Documents must follow certain rules.
- Data is structure is self-describing and extendable.

XML schema

XML Schema Examples - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xmlsdk/html/278f540d-b2c1-4907-bb49-2a659b17451f.asp

msdn

MSDN Home | Developer Centers | Library | Downloads | How to Buy | Subscribers | Worldwide

Search for

MSDN Library Search

Advanced Search

sync toc

- Windows Versions
- XML
  - Soap Toolkit
  - MSXML
    - MSXML SDK
      - MSXML
      - XML Standards Reference
        - Document Map
          - Elements [XML Standards]
          - Prolog [XML Standards]
          - XML Declaration [XML Standards]
          - Processing Instructions [XML Standards]
          - DOCTYPE Declaration [XML Standards]
          - XML Comments [XML Standards]
          - Textual Content [XML Standards]
          - Character and Entity References [XML Standards]
          - CDATA Sections [XML Standards]
          - Attributes [XML Standards]
          - White Space [XML Standards]
        - DTD Statements
          - ELEMENT
          - ATTLIST (Attribute List)
          - ENTITY
          - NOTATION
        - XML Schemas (XSD) Reference
          - XML Schema Examples
          - XML Schema Elements
            - <xsd:all> Element
            - <xsd:annotation> Element
            - <xsd:any> Element

This topic contains the World Wide Web Consortium (W3C) purchase order examples. The first example is the schema for the purchase order. The schema example.

Example: Purchase Order Schema

The following example shows a schema, po.xsd, that defines a purchase order. This example shows the use of **element**, and **attribute** declara **complexType** definitions.

Xml

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://tempuri.org/po.xsd"
xmlns="http://tempuri.org/po.xsd" elementFormDefault="qualified">
 <xs:annotation>
  <xs:documentation xml:lang="en">
   Purchase order schema for Example.com.
   Copyright 2000 Example.com. All rights reserved.
  </xs:documentation>
 </xs:annotation>

 <xs:element name="purchaseOrder" type="PurchaseOrderType"/>

 <xs:element name="comment" type="xs:string"/>

 <xs:complexType name="PurchaseOrderType">
  <xs:sequence>
   <xs:element name="shipTo" type="USAddress"/>
   <xs:element name="billTo" type="USAddress"/>
   <xs:element ref="comment" minOccurs="0"/>
   <xs:element name="items"  type="Items"/>
  </xs:sequence>
  <xs:attribute name="orderDate" type="xs:date"/>
 </xs:complexType>

 <xs:complexType name="USAddress">
  <xs:sequence>
   <xs:element name="name"   type="xs:string"/>
   <xs:element name="street" type="xs:string"/>
   <xs:element name="city"   type="xs:string"/>
   <xs:element name="state"  type="xs:string"/>
   <xs:element name="zip"    type="xs:decimal"/>
  </xs:sequence>
  <xs:attribute name="country" type="xs:NMTOKEN"
     fixed="US"/>
```

XLS

An XSL stylesheet contains some number of templates that define how the elements in an XML document should be transformed

```
xml version='1.0'?>
<authors>
  <author>
    <name>Mike Galos</name>
    <nationality>French</nationality>
  </author>
  <author period="modern">
    <name>Eva Corets</name>
    <nationality>British</nationality>
  </author>
  <author>
    <name>Cynthia Randall</name>
    <nationality>Canadian</nationality>
  </author>
  <author>
    <name>Stefan Knorr</name>
    <nationality>Canadian</nationality>
  </author>
  <author period="modern">
    <name>Paula Thurman</name>
    <nationality>British</nationality>
  </author>
</authors>
```

XLS transformation #1

Address J:\xml\authors.xml Go Links

```
  <?xml version="1.0" ?>
- <authors>
  - <author>
      <name>Mike Galos</name>
      <nationality>French</nationality>
    </author>
  - <author period="modern">
      <name>Eva Corets</name>
      <nationality>British</nationality>
    </author>
  - <author>
      <name>Cynthia Randall</name>
      <nationality>Canadian</nationality>
    </author>
  - <author>
      <name>Stefan Knorr</name>
      <nationality>Canadian</nationality>
    </author>
  - <author period="modern">
      <name>Paula Thurman</name>
      <nationality>British</nationality>
    </author>
  </authors>
```

XLS transformation #2

Address C:\xml\test\xsl_test.html Go Links

```
<authors>
  <author>
    <name>Mike Galos</name>
    <nationality>French</nationality>
  </author>
  <author period="modern" >
    <name>Eva Corets</name>
    <nationality>British</nationality>
  </author>
  <author>
    <name>Cynthia Randall</name>
    <nationality>Canadian</nationality>
  </author>
  <author>
    <name>Stefan Knorr</name>
    <nationality>Canadian</nationality>
  </author>
  <author period="modern" >
    <name>Paula Thurman</name>
    <nationality>British</nationality>
  </author>
</authors>
```

Simple XML example

Root element

Attribute

Empty tag

Attribute value

XML declaration

Closing tag

Untitled - Notepad

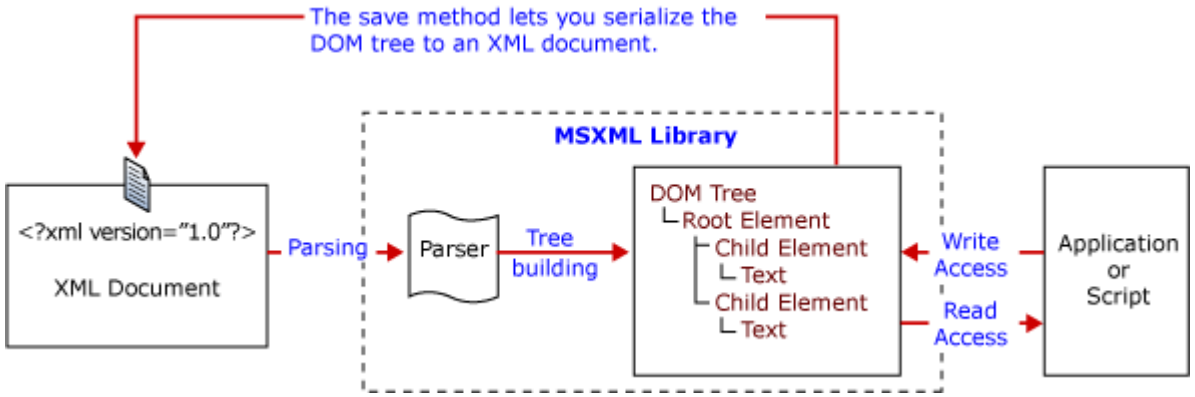
File Edit Format Help

```
<?xml version="1.0" standalone="yes"?>
<Recipes>
 <Recipe cook="The XML Gourmet">
  <Title>Bean Burrito</Title>
  <Category name="tex-mex"/>
  <Ingredients>
   <Item>1 can refried beans</Item>
   <Item>1 cup longhorn colby cheese, shredded</Item>
   <Item>1 small onion, finely chopped</Item>
   <Item>3 flour tortillas</Item>
  </Ingredients>
  <CookingInstructions>
   <Cinst>Empty can of refried beans into medium saucepan.</Cinst>
   <Cinst>Heat over medium-high heat until beans are smooth and bubbly.</Cinst>
   <Cinst>Warm tortillas in microwave for 30 seconds.</Cinst>
  </CookingInstructions>
  <ServingInstructions>
   <Sinst>Spread 1/3 of warm beans on each tortilla.</Sinst>
   <Sinst>Sprinkle with cheese and onions.</Sinst>
   <Sinst>Roll tortillas and serve.</Sinst>
  </ServingInstructions>
 </Recipe>
</Recipes>
```

Root element

Opening tag

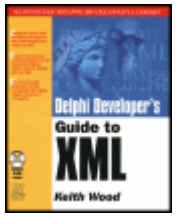
DOM with MSXML

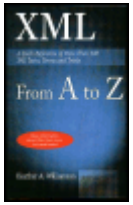


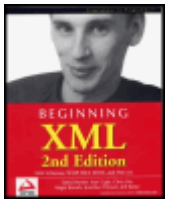
The Document Object Model (DOM) as implemented in MSXML provides a programmatic representation of XML documents.

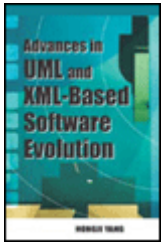
XML Resources

Amazon.com over 980 books about XML !

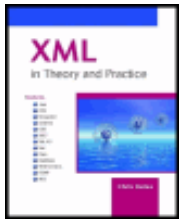








Reference point XML

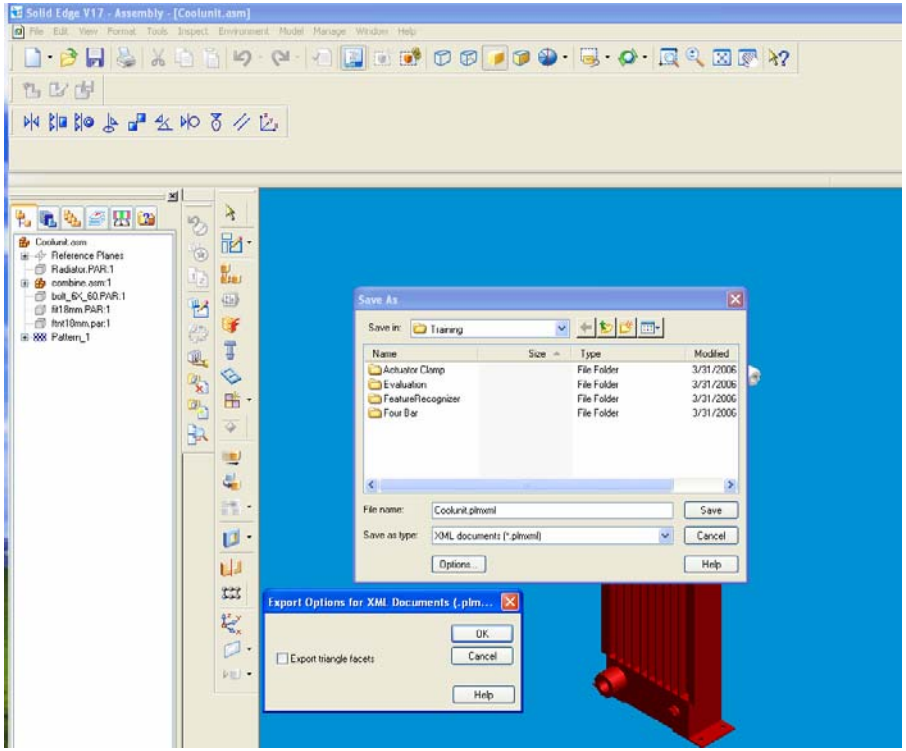


PLMXML Schema

```
<?xml version="1.0" standalone="no"?>
<PLMXML xmlns "http://www.plmxml.org/Schemas/PLMXMLSchema" schema
 xmlns:xsi="h
time="14:49:12" language="en" date="2001-12-19">
  <ProductDef id="id1" name="assembly" revision="1">
    <InstanceGraph id="id2" rootInstanceRef="id3">
      <Description>
        Product hierachy for sample car, distributed over two fil
      </Description>
      <Instance id="id3" partRef="#id4"/>
      <Instance id="id6" partRef="#id7"/>
      <Instance id="id10" partRef="car_axle_assembly.plmxml#id11"
      <Instance id="id24" partRef="car_axle_assembly.plmxml#id11"
      <Part id="id4" type="assembly" instanceRefs="id6 id10 id24"
      </Part>
      <Part id="id7" type="solid" representationRefs="id9">
      </Part>
      <Transform id="id23">1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1</Trans
      <Transform id="id25">1 0 0 0 0 1 0 0 0 0 1 0 0 -1 0 1</Tran
    </InstanceGraph>
    <Representation id="id9" name="car_id7" format="XT" revision=
      <Description>
        Chassis of Car assembly
      </Description>
    </Representation>
    <ProductView id="id26" primaryOccurrenceRef="id27"
```

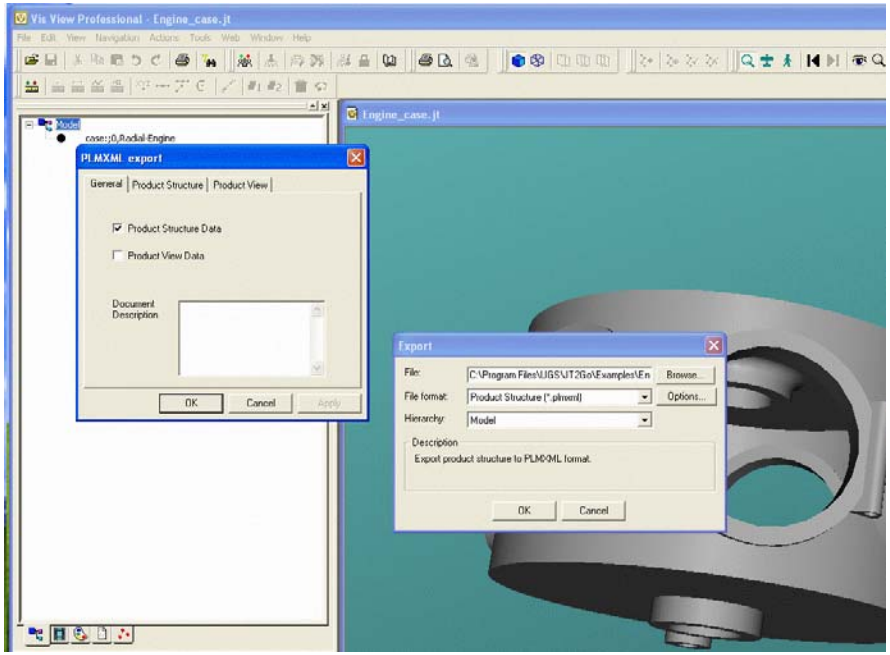
PLM XML is defined by a set of XML schemas that are compliant with the W3C XML schema recommendation.

Creating PLMXML



With SolidEdge

With VisView Professional



Loading PLMXML

Example code for loading PLMXML file with JavaScript.

```
source = new ActiveXObject("Msxml2.DOMDocument.5.0");

  source.async=false;
  source.load(“myNEW.plmxml);
  source.setProperty("SelectionLanguage", "XPath");
 if (source.parseError.errorCode != 0)
  {
    result = reportParseError(source.parseError);
    alert(result);
  }
```

Transforming with XSLT

Example code for PLMXML transformation file with JavaScript.

```
stylesheet = new ActiveXObject("Msxml2.DOMDocument.5.0");
stylesheet.async = false;
stylesheet.load("hilite-xml.xsl");

if (source.parseError.errorCode != 0) {
      alert( reportParseError(source.parseError));
   }

  try {
    result = source.transformNode(stylesheet);
  }
  catch (e) {
    alert( reportRuntimeError(e));
  }
```

Query with XPath

```
function doQuery(theQuery) {
    var q = source.selectNodes(theQuery);
    alert(q.length + " Objects Found")
    for (var e = q.nextNode(); e != null; e = q.nextNode() {
      try {
              alert( "Node "   + e.xml )
      }
      catch (exception) {
          alert reportRuntimeError(exception);
     }
    }
  }
```

theQuery="PLMXML/ProductDef/InstanceGraph/Instance "
theQuery=""PLMXML/ProductDef/Part "
theQuery=""PLMXML/ProductDef/InstanceGraph/Transform"
theQuery=""PLMXML/ProductDef/Part/UserData/UserValue[@title='ProjectName']/@value"
theQuery=""PLMXML/ProductDef/InstanceGraph/Transform"

PLMXML XPath Expression Tester

PLMXML DEMO INTERFACE - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address C:\xml\presentation\Copy of VIS_xml_Test-plm-patterns.htm

Demonstration of quering against an PLMXML document using XPath expressions.

Input File Name:

C:\interop\XML\modTest.plmxml [Browse...] [Show Selected]

XPath Expression:

PLMXML/ProductDef/InstanceGraph/Instance [Highlite Nodes]

Some Sample Queries:

PLMXML/ProductDef/InstanceGraph/Instance
PLMXML/ProductDef/InstanceGraph/Part
PLMXML/ProductDef/Part
PLMXML/ProductDef/Part/UserData/UserValue[@title='ProjectName']/@value
PLMXML/ProductDef/InstanceGraph/Transform
PLMXML/ProductDef/Part/UserData/UserValue[@title='PartNumber']/@value
PLMXML/ProductDef/Part/Representation/CompoundRep/@location

[Select part Number]

PLMXML source document


```
<PLMXML schemaVersion="4" date="2006-04-12" time="18:51:14" author="SE Adapter 17.0.0.65" >
  <Description>plmxmlAdapterSE</Description>
  <ProductDef id="id1" name="C:\Program Files\Solid Edge V17\Training\Seracvs.asm" revision="1" >
    <InstanceGraph id="id2" rootInstanceRef="id4" >
      <Instance id="id4" name="C:\Program Files\Solid Edge V17\Training\Seracvs.asm" partRef="#thisAsm" />
      <Instance id="id713" name="Housing Base.par:1" partRef="#id8" transformRef="id714" >
        <ApplicationRef version="23481" application="SE" label="AttachmentID 106" />
      </Instance>
      <Instance id="id717" name="Housing Top.par:1" partRef="#id56" transformRef="id718" >
        <ApplicationRef version="12614" application="SE" label="AttachmentID 210" />
      </Instance>
      <Instance id="id721" name="Heater Plate.psm:1" partRef="#id75" transformRef="id722" >
        <ApplicationRef version="1221" application="SE" label="AttachmentID 1504" />
      </Instance>
      <Instance id="id725" name="spout.asm:1" partRef="#id94" transformRef="id726" >
        <ApplicationRef version="8115" application="SE" label="AttachmentID 660" />
      </Instance>
      <Instance id="id729" name="Spout Top.par:1" partRef="#id223" transformRef="id730" >
        <ApplicationRef version="12334" application="SE" label="AttachmentID 145" />
      </Instance>
      <Instance id="id733" name="Spout Bottom.par:1" partRef="#id268" transformRef="id734" >
        <ApplicationRef version="22165" application="SE" label="AttachmentID 141" />
      </Instance>
      <Instance id="id737" name="Coffee Pot.asm:1" partRef="#id96" transformRef="id738" >
        <ApplicationRef version="1122" application="SE" label="AttachmentID 611" />
      </Instance>
      <Instance id="id741" name="strainer.asm:1" partRef="#id300" transformRef="id742" >
        <ApplicationRef version="19405" application="SE" label="AttachmentID 230" />
      </Instance>
```

Traversing PLMXML Nodes

```
nodes=source.getElementsByTagName("ProductDef");

parts = nodes[0].getElementsByTagName("Part");

for ( var i=0; I <parts.length; i++) {
  ud=parts[i].getElementsByTagName("UserData");

  uv =ud.getElementsByTagName ("UserValue");
  parts[i].getElementsByTagName("Representation");

  rep=parts[i].getElementsByTagName("CompoundRep");

  rep.getElementsByTagName("MassProperties");

}
```


```
<?xml version="1.0" ?>
- <PLMXML schemaVersion="4" date="2006-04-12" time="18:51:14" author="SE Adapter 17.0.0.65">
  <Description>plmxmlAdapterSE</Description>
  <ProductDef id="id1" name="C:\Program Files\Solid Edge V17\Training\Seracvs.asm" revision="1"
  + <InstanceGraph id="id2" rootInstanceRef="id4" />
  - <Part id="id8" name="C:/Program Files/Solid Edge V17/Training/Housing Base.par"
      preferredUnits="inches" type="solid">
      <ApplicationRef version="23481" application="SE" label="C:/Program Files/Solid Edge
        V17/Training/Housing Base.par" />
    - <UserData id="id14" type="project">
        <Description>Project Properties</Description>
        <UserValue value="" title="ProjectName" />
        <UserValue value="009J73746" title="PartNumber" />
      </UserData>
      <Representation id="id10" name="Main" format="XT" equivalentRef="id11">
      - <UserData id="id16" type="data">
          <UserValue value="Main" title="refset" />
        </UserData>
      - <CompoundRep id="id19" name="DETAILED 67" format="XT" location="Housing%
          20Base_XT_Body_67_-1_xt.xmt_txt" load="true">
        - <UserData id="id21" type="data">
            <UserValue value="part" title="usage" />
          </UserData>
        - <MassProperties id="id23">
            <Density>0</Density>
          </MassProperties>
        </CompoundRep>
      </Representation>
    - <Representation id="id11" name="Main" format="jXT" equivalentRef="id10">
      - <UserData id="id18" type="data">
          <UserValue value="Main" title="refset" />
        </UserData>
      - <CompoundRep id="id24" name="DETAILED 67" format="jXT" location="Housing%
          20Base_XT_Body_67_-1_jxt.jxt" load="true">
        - <UserData id="id26" type="data">
            <UserValue value="part" title="usage" />
          </UserData>
        </CompoundRep>
      </Representation>
    </Part>
  + <Part id="id56" name="C:/Program Files/Solid Edge V17/Training/Housing Top.par"
      preferredUnits="inches" type="solid">
  + <Part id="id75" name="C:/Program Files/Solid Edge V17/Training/Heater Plate.psm"
      preferredUnits="inches" type="solid">
  + <Part id="id100" name="C:/Program Files/Solid Edge V17/Training/Housing Center.par"
      preferredUnits="inches" type="solid">
```

Fetching PLMXML Nodes Attributes

```
partID=part[i].getAttribute("id");
partTYPE= part[i]=itemsList[i].getAttribute("type");
…
partNUMBER=ud[1]. getAttribute(“PartNumber”);
```


```
<?xml version="1.0" ?>
- <PLMXML schemaVersion="4" date="2006-04-12" time="18:51:14" author="SE Adapter 17.0.0.65">
  <Description>plmxmlAdapterSE</Description>
  - <ProductDef id="id1" name="C:\Program Files\Solid Edge V17\Training\Seracvs.asm" revision="1">
    + <InstanceGraph id="id2" rootInstanceRef="id4">
    - <Part id="id8" name="C:/Program Files/Solid Edge V17/Training/Housing Base.par"
        preferredUnits="inches" type="solid">
        <Application Ref version="23481" application="SE" label="C:/Program Files/Solid Edge
          V17/Training/Housing Base.par" />
      - <UserData id="id14" type="project">
          <Description>Project Properties</Description>
          <UserValue value="" title="ProjectName" />
          <UserValue value="009J73746" title="PartNumber" />
        </UserData>
      - <Representation id="id10" name="Main" format="XT" equivalentRef="id11">
        - <UserData id="id16" type="data">
            <UserValue value="Main" title="refset" />
          </UserData>
        - <CompoundRep id="id19" name="DETAILED 67" format="XT" location="Housing%
            20Base_XT_Body_67_-1_xt.xmt_txt" load="true">
          - <UserData id="id21" type="data">
              <UserValue value="part" title="usage" />
            </UserData>
          - <MassProperties id="id23">
              <Density>0</Density>
            </MassProperties>
          </CompoundRep>
        </Representation>
      - <Representation id="id11" name="Main" format="jXT" equivalentRef="id10">
        - <UserData id="id18" type="data">
            <UserValue value="Main" title="refset" />
          </UserData>
        - <CompoundRep id="id24" name="DETAILED 67" format="jXT" location="Housing%
            20Base_XT_Body_67_-1_jxt.jxt" load="true">
          - <UserData id="id26" type="data">
              <UserValue value="part" title="usage" />
            </UserData>
          </CompoundRep>
        </Representation>
      </Part>
    + <Part id="id56" name="C:/Program Files/Solid Edge V17/Training/Housing Top.par"
        preferredUnits="inches" type="solid">
    + <Part id="id75" name="C:/Program Files/Solid Edge V17/Training/Heater Plate.psm"
        preferredUnits="inches" type="solid">
    + <Part id="id100" name="C:/Program Files/Solid Edge V17/Training/Housing Center.par"
        preferredUnits="inches" type="solid">
```

Questions ?