EMEA Tivoli & Security Technical Conference 2012

IBM Tivoli Directory Integrator – Web Services Integration

SEC16

Franz Wolfhagen (franzw@dk.ibm.com)
IBM Certified IT Specialist

The Web Services Integration Challenge
Web Services Patterns

Web Services

- Web Services are used to support a Service Oriented Architecture (SOA) to enhance the efficiency, agility and productivity of an enterprise, by exposing business processes as reusable services
- Services can be exposed through widely available mechanisms and protocols such as SOAP/HTTP(S) and SOAP/JMS
- Many enterprises uses Message Broker systems to provide an Enterprise Service Bus (ESB)

What is SOAP?

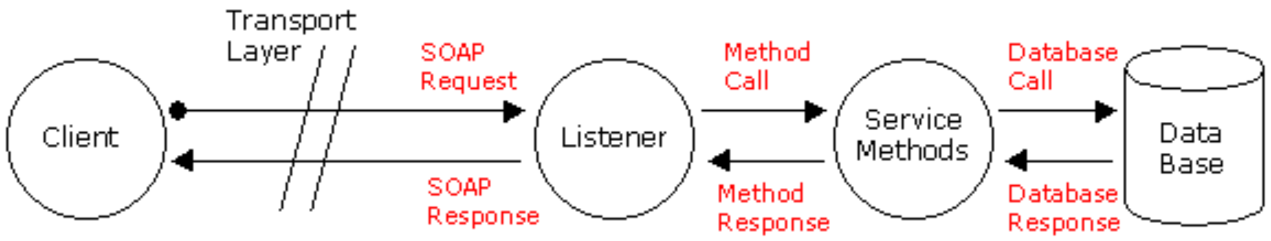
- SOAP is an XML-based messaging protocol
- Defines a set of rules for structuring messages that can be used for simple one-way messaging or performing RPC-style (Remote Procedure Call) request-response dialogues
- Not tied to any transport protocol though HTTP is popular (JMS,SMTP are other examples)
- SOAP may wrap other XML data frameworks e.g. SPML,SAML and XMLDsig

Other Web Service Protocols

- RPC (Remote Procedure Call)
  - Call function with argument
  - XML-RPC (XML body format)
  - JSON-RPC (JSON body format)
- REST (REpresentional State Transfer)
  - Generally HTTP(s) protocol
  - URLs are ressource locations
  - Response is delivered via status codes

Simple Web Services Patterns

- A Simple request/response may look like this



- Security can be provided by the the protocol e.g. Simple Authentication over HTTPS
- Request/Response over HTTP(S) is session based
- Request/Response over JMS can be performed via correlation ID retrieval/lookup or other JMS Header

Complex Web Service Patterns

- Some Web Services may seperate Authentication in a seperate step e.g. :
  - Start session – a challenge is returned
  - Sign the challenge with your PKCS12 Certificate
  - Return signed challenge – a session token is returned
  - Call a functional service (session token used for Authentication)

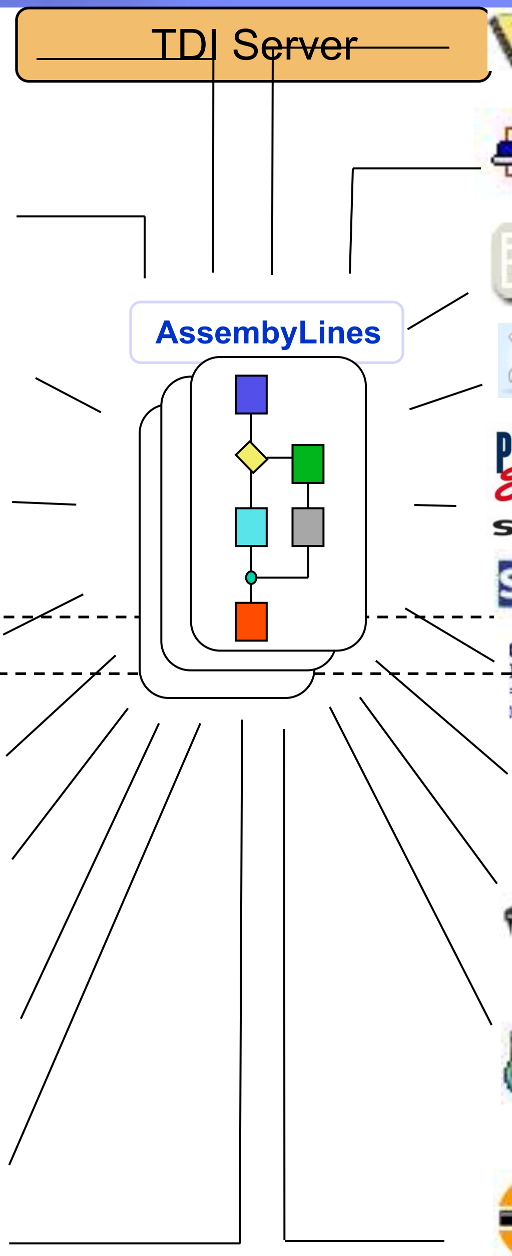
Complex Web Service Patterns cont.

- Web Services over JMS may deliver multiple responses e.g. :
  - The Service Client puts a Service request on the request queue and retrieves the correlation ID
  - The Service Server creates a number of XML documents in response that is posted to the response queue with the common correlation ID
  - The Service Client pulls all responses with the correlation ID off the queue
- NOTE : In such a pattern completeness checks are part of the message and ensure by the transport protocol.

ITDI Out-Of-the-Box Support for Web Services

Why use ITDI for Web Services Integration ?

The ”Gaffer Tape” of IT Integration



Web Service Server
Web Service Invoke
Axis Java-to-Soap
Axis Soap-to-Java

LDAP
LDAP Server
Tivoli Access Manager
Windows Users and Groups

Active Directory Changelog
IBM Directory Server Changelog
Netscape/iPlanet Changelog
zOS LDAP Changelog

JDBC
TDI Properties
SystemStore
RDBMS Changelog

AssemblyLine
Server Notifications
AssemblyLine Function Component

Domino Change Detection
Domino Users
Lotus Notes

Exchange Changelog
Email (POP, IMAP)
SendEMail Function

TIM DSMLv2
DSMLv2 SOAP
DSML v2 SOAP Server
Generic JNDI
ITIM Agent

EMF SDOToXML Function
EMF XMLToSDO Function

Timer

Generic Log Adapter
RAC
CBE - CommonBaseEvent components

JMX
SNMP
SNMP Server
TCP
TCP Server

Remedy/Peregrine /CCMDB tickets
Maximo MEA/IF/Tpae
Many custom Components on
OPAL, tdi-users.org or on request

CCMDB
Netcool
RSS
Google

PeopleSoft
Siebel
SAP ALE Idoc
SAP R/3 Business Object Repository
SAP R/3 User Registry
SAP R/3 RFC Component

Script Connector
Generic Java Method
Parser FC

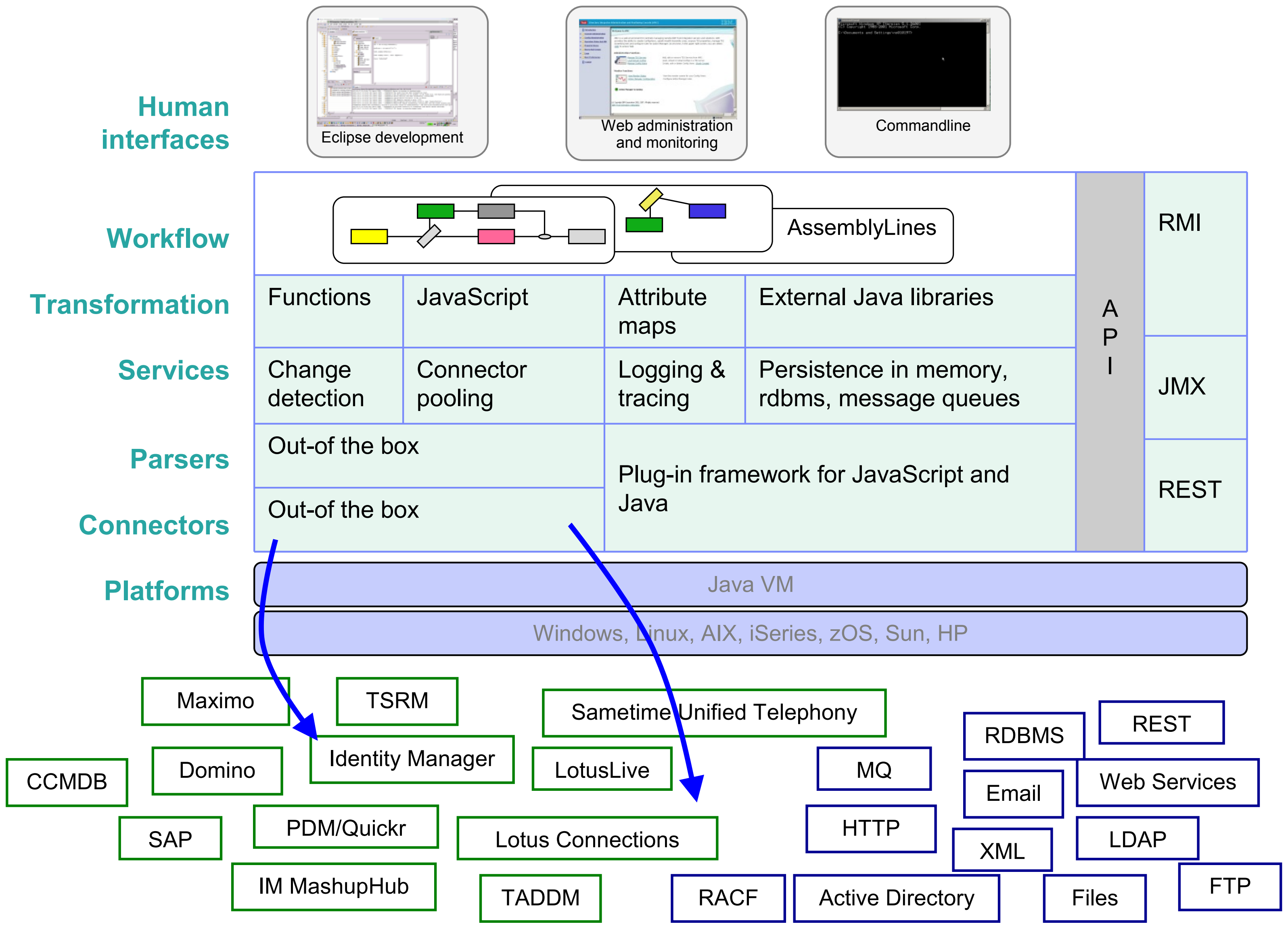
Remote secure command Line
z/OS TSO/E Command Line
Command Line

MemQueue
Memory Stream

HTTP/REST
File System
FTP Client
URL
HTTP Client
HTTP Server

IBM MQ Series
JMS Pub/Sub
MQe Password Store

Parsers
CSV
DSML v1 & v2
Fixed Record
HTTP
LDIF
Line reader/writer
SOAP
Script
XML
XML Sax
XSL


Human interfaces
Eclipse development
Web administration and monitoring
Commandline
Workflow
AssemblyLines
Transformation
Functions
JavaScript
Attribute maps
External Java libraries
Services
Change detection
Connector pooling
Logging & tracing
Persistence in memory, rdbms, message queues
Parsers
Out-of the box
Connectors
Out-of the box
Plug-in framework for JavaScript and Java
API
RMI
JMX
REST
Platforms
Java VM
Windows, Linux, AIX, iSeries, zOS, Sun, HP
Maximo
TSRM
Sametime Unified Telephony
RDBMS
REST
CCMDB
Domino
Identity Manager
LotusLive
MQ
Email
Web Services
SAP
PDM/Quickr
Lotus Connections
HTTP
XML
LDAP
IM MashupHub
TADDM
RACF
Active Directory
Files
FTP

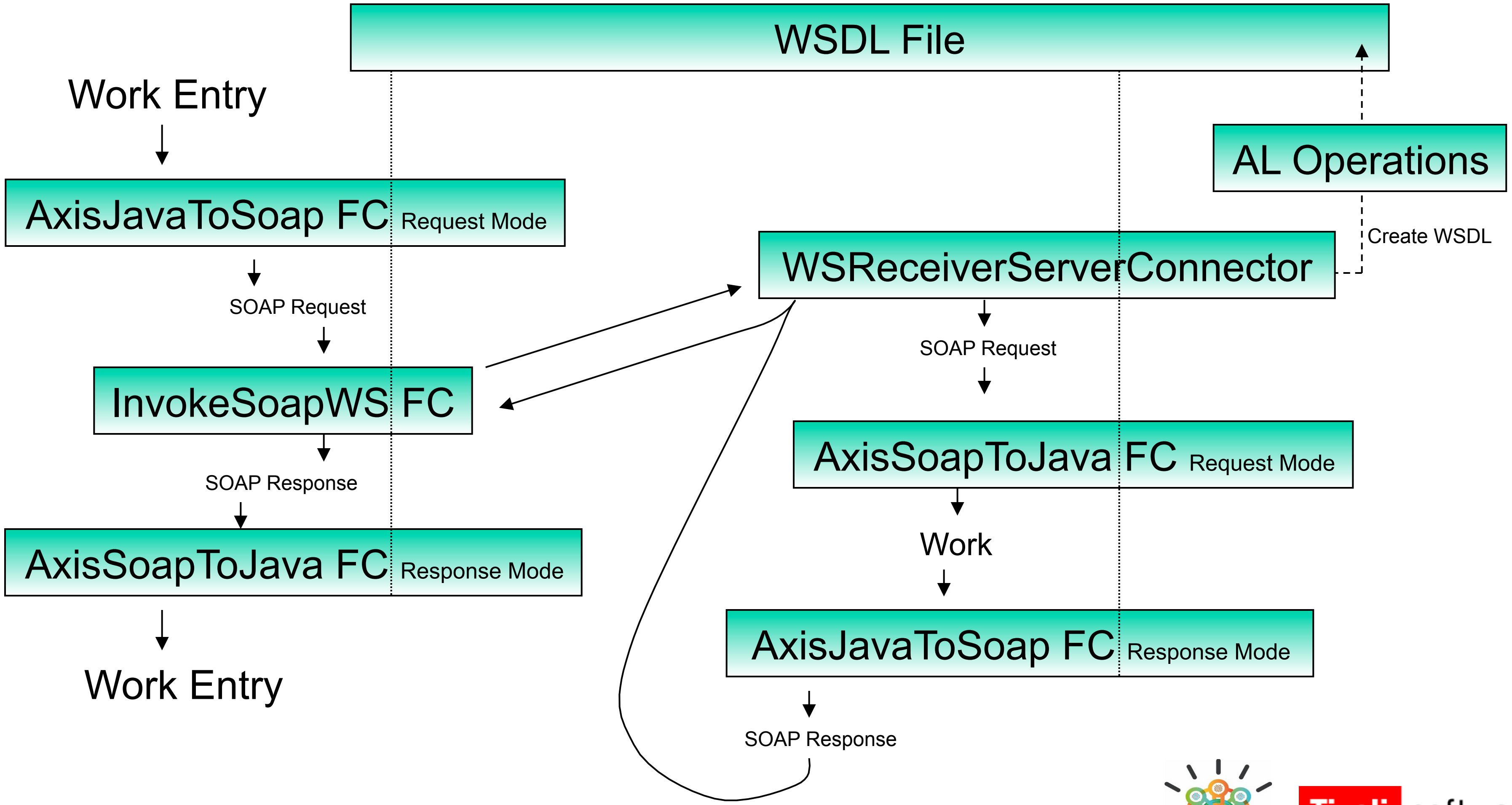
ITDI Web Service Components

- Connectors
  - Axis Easy Web Service Server (WS Server)
  - Axis2 Web Service Server (WS server)
  - DSMLv2 SOAP Connector / Server Connector (client/ server)
- Parsers
  - DSMLv1/v2
  - JSON
  - SOAP
  - XML parsers (Simple/XML/SAX/XSL Based)

ITDI Web Service Components cont.

- Function Components
  - Castor Java <-> XML (complex/custom data types)
  - WrapSoap
  - InvokeSoap
  - Axis Soap <-> Java (serializer/deserializer)
  - Axis2 WS Client
  - Axis EasyInvoke Soap
  - Complex Types Generator (for the Axis Soap FC)

Using Key ITDI Web Service Components


WSDL File
Work Entry
AxisJavaToSoap FC Request Mode
SOAP Request
InvokeSoapWS FC
SOAP Response
AxisSoapToJava FC Response Mode
Work Entry
AL Operations
Create WSDL
WSReceiverServerConnector
SOAP Request
AxisSoapToJava FC Request Mode
Work
AxisJavaToSoap FC Response Mode
SOAP Response

Building/Parsing the XML Data
How to Manage XML Data in ITDI

XML

- eXtensible Markup Language
- Metalanguage - used to create other languages
- Has become a universal data-exchange format

- You need to ensure that XML is
  - Well-Formed: Structure follows XML syntax rules
  - Valid: Structure conforms to a Schema

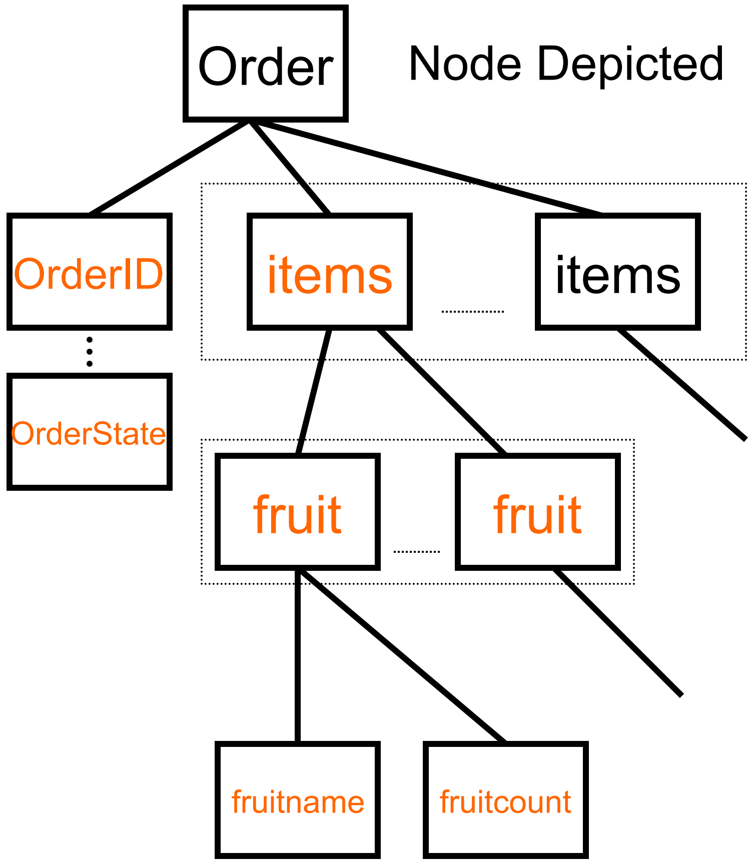
What is a XML Node ?

- An XML node is a tree, containing an open tag, contents, and a close tag
  - <foo id="123">This is <bar>an element</bar></foo>
  - Here, the tag named 'foo' encloses the contents and attributes of the Node
  - In this case node with the tag 'foo' contains a node with the tag 'bar' also.

There is another notion used in XML literature called Element, that is skipped here to keep things simple

XML is a Nested Tree

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<IBM>
 <eFruit>
  <FruitSales>
        <Order>
         <OrderId OrderState="ReadOnly">E0001DU9</OrderId>
         <Items>
            <Fruit>
               <FruitName>Grapes</FruitName>
               <FruitCount>12</FruitCount>
             </Fruit>
             <Fruit>
               <FruitName>Apples</FruitName>
               <FruitCount>3</FruitCount>
             </Fruit>
         </Items>
        </Order>
  </FruitSales>
 </eFruit>
</IBM>
```



attribute & value

element & content

Nodes

Branch nodes contain children

Leaf nodes contain content

Attributes, Values, etc.

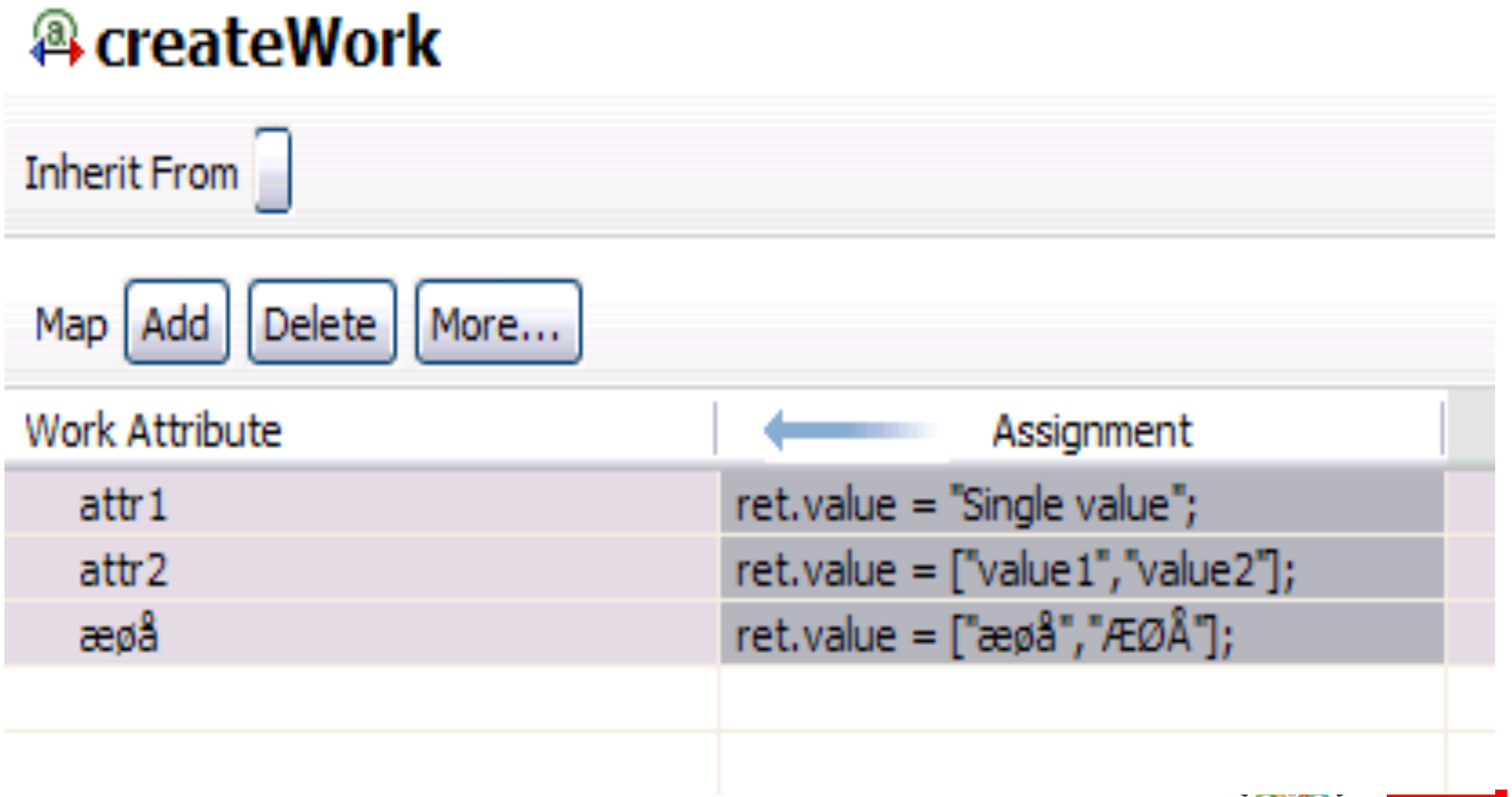
XML Syntax

- Tags properly nested
- Tag names case-sensitive
- All tags must be closed
  - or self-closing
  - <foo/> is the same as <foo></foo>
- Attributes enclosed in quotes
- Document consists of a single (root) element

Building XML Using a ”scripted” Parser

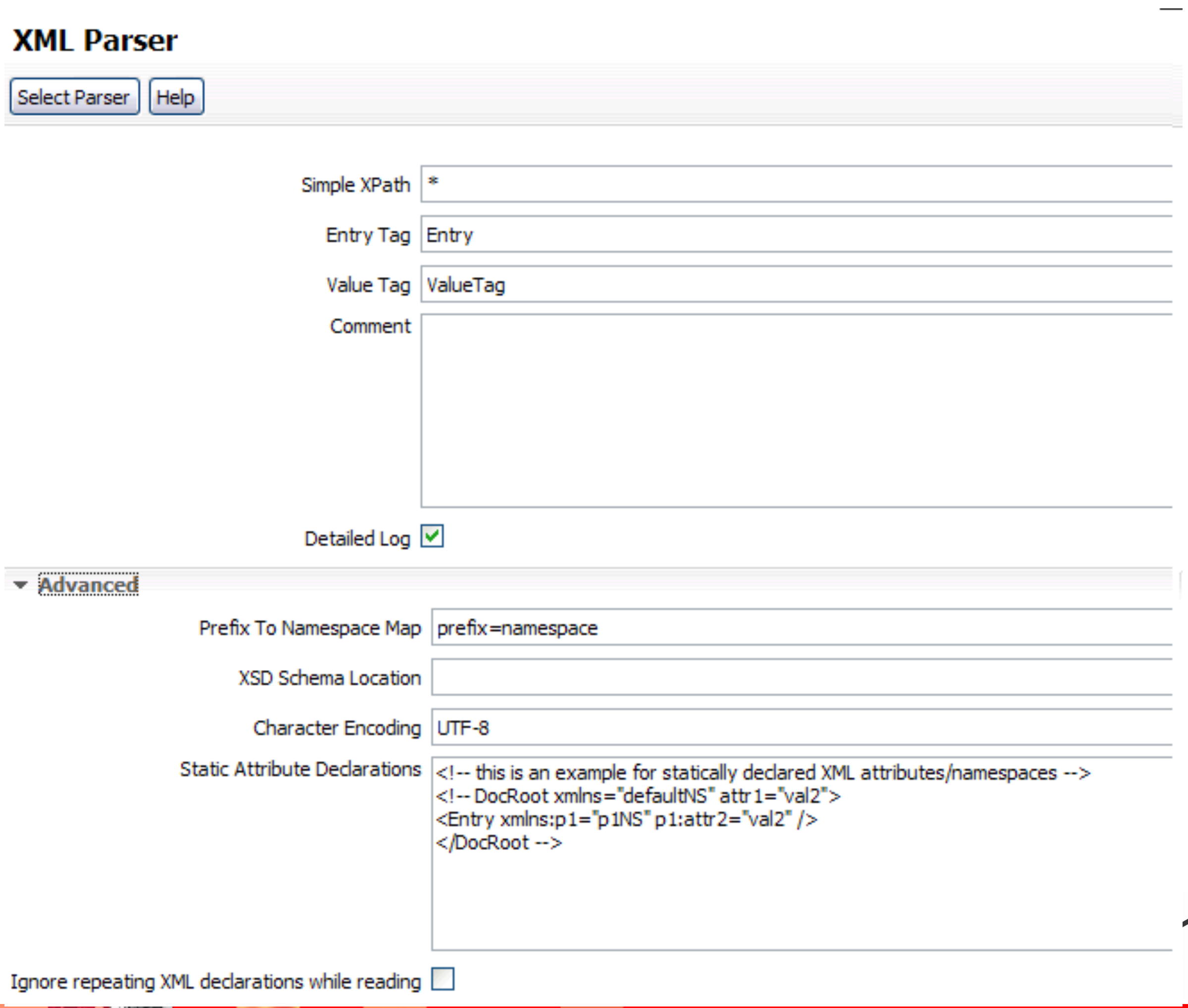
```
//Setup the parser from the Parsers Ressource
var myParser = system.getParser("Parsers/someParser");

//Check if it worked
if (myParser == null) throw "Unable to get Parsers/someParser";

//Create a Java Outputstream and connect it the parser
os = new java.io.ByteArrayOutputStream();
myParser.setOutputStream(os);

//Initialize the parser
myParser.initParser();

//Write an entry to the parser and close it
myParser.writeEntry(someEntry);
myParser.closeParser();

//The parsed entry is now available in the outputstream
task.logmsg(”Result : ” + os.toString(”UTF-8”));
```

Converting an Entry to XML
The Work Entry



Work Attribute	Assignment
attr1	ret.value = "Single value";
attr2	ret.value = ["value1","value2"];
æøå	ret.value = ["æøå","ÆØÅ"];

Converting an Entry to XML Using the (StAX) XML Parser



Simple XPath	*
Entry Tag	Entry
Value Tag	ValueTag
Comment	
Detailed Log	✓
Prefix To Namespace Map	prefix=namespace
XSD Schema Location	
Character Encoding	UTF-8
Ignore repeating XML declarations while reading	☐

Static Attribute Declarations

```
<!-- this is an example for statically declared XML attributes/namespaces -->
<!-- DocRoot xmlns="defaultNS" attr1="val2">
<Entry xmlns:p1="p1NS" p1:attr2="val2" />
</DocRoot -->
```

Converting an Entry to XML Using the (StAX) XML Parser

```
var myEntry = work;
//Setup the StAX XML parser
var xml2 = system.getParser("Parsers/XML2");
if (xml2 == null) throw "Unable to get Parsers/
     XML2";
os = new java.io.ByteArrayOutputStream();
xml2.setOutputStream(os);
xml2.initParser();
xml2.writeEntry(myEntry);
xml2.closeParser();
task.logmsg("This is the work entry as parsed with
     the (StAX) XML parser : \n" +
     os.toString("UTF-8").trim() + "\n");
```

```
INFO  - This is the work entry as parsed with the (StAX)
XML parser :

<DocRoot>
  <Entry>
    <attr2>
      <ValueTag>value1</ValueTag>
      <ValueTag>value2</ValueTag>
    </attr2>
    <attr1>Single value</attr1>
    <æøå>
      <ValueTag>æøå</ValueTag>
      <ValueTag>ÆØÅ</ValueTag>
    </æøå>
  </Entry>
</DocRoot>
```

Building XML Data Directly in ITDI

- With ITDI Version 7.0 XML became an integrated part of ITDI and the Entry object is DOM enabled by default
- Hence an XML can be built very easily :

```
var myEntry = system.newEntry();

myEntry["attr1"] = "Single value";

//Using Array syntax
myEntry["attr2"] = null;
myEntry["attr2"][0] = "value1";
myEntry["attr2"][1] = "value2";

//Using space seperated
myEntry["æøå"] = "æøå ÆØÅ";

task.logmsg("\n" + myEntry.toXML());
```

```
Output from the script :


<attr1>Single value</attr1>
<attr2>
value1
value2
</attr2>
<æøå>æøå ÆØÅ</æøå>
```

- Note – the XML is not really valid (multirootet)

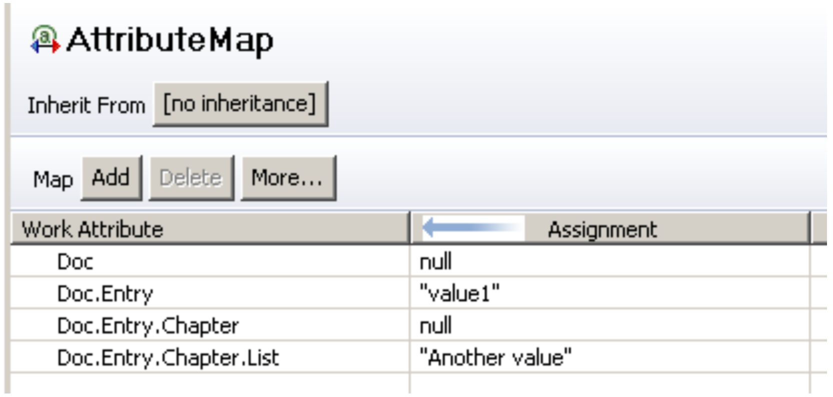
Building Multilevel XML with Attributes

```
var myEntry = system.newEntry();


//Toplevel
myEntry["Doc"] = null;


//Adding an aattribute
myEntry["Doc"]["@id"] = "my_id";


//Adding second level with a value
myEntry["Doc"]["Entry"] = "value1";


//Building more levels
myEntry["Doc"]["Entry"]["Chapter"] = null;
myEntry["Doc"]["Entry"]["Chapter"]["List"] =
"Another value";


task.logmsg("\n" + myEntry.toXML());
```

```
<Doc id="my_id">
  <Entry>
    value1
    <Chapter>
      <List>Another value</List>
    </Chapter>
  </Entry>
</Doc>
```

Building Multilevel XML with Attributes Alternative Way with Attributemap

- Note: Attributes are NOT supported using this simple syntax



Work Attribute	Assignment
Doc	null
Doc.Entry	"value1"
Doc.Entry.Chapter	null
Doc.Entry.Chapter.List	"Another value"

```
<Doc>
        <Entry>
                value1
                <Chapter>
                        <List>Another value</List>
                </Chapter>
        </Entry>
</Doc>
```

Building Multinode XML

```
xmlEntry = system.newEntry();


xmlEntry["Doc"] = null;
//parent = xmlEntry["Doc"];
//myChapter = system.newAttribute("Chapter");
//parent.appendChild(myChapter);


//above is equal to :
xmlEntry["Doc.Chapter"] = "no 1";


//Adding another Chapter
parent = xmlEntry["Doc"];
myChapter = system.newAttribute("Chapter");
myChapter.addValue("no 2");
parent.appendChild(myChapter);


task.logmsg("\n" + xmlEntry.toXML());
```

```
<Doc>
     <Chapter>no 1</Chapter>
     <Chapter>no 2</Chapter>
</Doc>
```

Building XML with Namespaces

- The Target XML :

```
<ds:Object xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:openoces="http://www.openoces.org/2006/07/signature#" Id="ToBeSigned">
	<ds:SignatureProperties>
		<ds:SignatureProperty>
			<openoces:Name>host</openoces:Name>
			<openoces:Value Encoding="base64" VisibleToSigner="no">d3d3Lm5ldHMtZGFuaWQuZGs=</openoces:Value>
		</ds:SignatureProperty>
	</ds:SignatureProperties>
</ds:Object>
```

Building XML with namespaces cont.

- First try (Will be fixed in a later Fixpack):

```
xmlEntry = system.newEntry();
//Create Root Object
xmlEntry["ds:Object"]=null;
//Adding Namespaces
xmlEntry["ds:Object"]["@xmlns:ds"] = "http://www.w3.org/2000/09/xmldsig#";
xmlEntry["ds:Object"]["@xmlns:openoces"] = "http://www.openoces.org/2006/07/signature#";
xmlEntry["ds:Object"]["@Id"] = "ToBeSigned";

xmlEntry["ds:Object.ds:SignatureProperties.ds:SignatureProperty.openoces:Name"] = "host";
xmlEntry["ds:Object.ds:SignatureProperties.ds:SignatureProperty.openoces:Value"]
           = system.base64Encode(("www.nets-danid.dk").getBytes());
xmlEntry["ds:Object.ds:SignatureProperties.ds:SignatureProperty.openoces:Value"]["@Encoding"] = "base64";
xmlEntry["ds:Object.ds:SignatureProperties.ds:SignatureProperty.openoces:Value"]["@VisibleToSigner"] = "no";

task.logmsg("\n" + xmlEntry.toXML());
```

```
<Object xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:openoces="http://www.openoces.org/2006/07/signature#" Id="ToBeSigned">
        <ds:SignatureProperties>
                <ds:SignatureProperty>
                        <openoces:Name>host</openoces:Name>
                        <openoces:Value Encoding="base64" VisibleToSigner="no">d3d3Lm5ldHMtZGFuaWQuZGs=</openoces:Value>
                </ds:SignatureProperty>
        </ds:SignatureProperties>
</Object>
```

Building XML with namespaces cont.

- This works on current level:

```
xmlEntry = system.newEntry();
//Create Root Object with Namespace
var object = xmlEntry.createElementNS("http://www.w3.org/2000/09/xmldsig#","ds:Object");
xmlEntry.setAttribute(object);
//Adding Namespaces
xmlEntry["ds:Object"]["@xmlns:openoces"] = "http://www.openoces.org/2006/07/signature#";
xmlEntry["ds:Object"]["@Id"] = "ToBeSigned";

xmlEntry["ds:Object.ds:SignatureProperties.ds:SignatureProperty.openoces:Name"] = "host";
xmlEntry["ds:Object.ds:SignatureProperties.ds:SignatureProperty.openoces:Value"]
      = system.base64Encode(("www.nets-danid.dk").getBytes());
xmlEntry["ds:Object.ds:SignatureProperties.ds:SignatureProperty.openoces:Value"]["@Encoding"] = "base64";
xmlEntry["ds:Object.ds:SignatureProperties.ds:SignatureProperty.openoces:Value"]["@VisibleToSigner"] = "no";

task.logmsg("\n" + xmlEntry.toXML());
```

```
<ds:Object xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:openoces="http://www.openoces.org/2006/07/signature#" Id="ToBeSigned">
    <ds:SignatureProperties>
        <ds:SignatureProperty>
            <openoces:Name>host</openoces:Name>
            <openoces:Value Encoding="base64" VisibleToSigner="no">d3d3Lm5ldHMtZGFuaWQuZGs=</openoces:Value>
        </ds:SignatureProperty>
    </ds:SignatureProperties>
</ds:Object>
```

Reading the Response

A Soap Response Example - Complete

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <soap:Body>
                <SPMLSearchRequestResponse xmlns="http://www.kmd.dk/KMD.YH.KSPAabenSpml">
                        <SPMLSearchRequestResult>
                                &lt;spml:searchResponse xmlns:spml=&quot;urn:oasis:names:tc:SPML:1:0&quot; xmlns:dsml=&quot;urn:oasis:names:tc:DSML:2:0:core
result=&quot;urn:oasis:names:tc:SPML:1:0#success&quot;   &gt;
                                        &lt;searchResultEntry&gt;
                                                &lt;spml:identifier type=&quot;URN:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName&quot;&gt;
                                                        &lt;spml:id&gt;QAAF&lt;/spml:id&gt;
                                                &lt;/spml:identifier&gt;
                                                &lt;spml:attributes&gt;
                                                        &lt;dsml:attr name=&quot;uid&quot;&gt;
                                                                &lt;dsml:value&gt;QAAF&lt;/dsml:value&gt;
                                                        &lt;/dsml:attr&gt;
                                                &lt;/spml:attributes&gt;
                                        &lt;/searchResultEntry&gt;
                                                .........
                                        &lt;searchResultEntry&gt;
                                                &lt;spml:identifier type=&quot;URN:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName&quot;&gt;
                                                        &lt;spml:id&gt;T9UDK01&lt;/spml:id&gt;
                                                &lt;/spml:identifier&gt;
                                                &lt;spml:attributes&gt;
                                                        &lt;dsml:attr name=&quot;uid&quot;&gt;
                                                                &lt;dsml:value&gt;T9UDK01&lt;/dsml:value&gt;
                                                        &lt;/dsml:attr&gt;
                                                &lt;/spml:attributes&gt;
                                        &lt;/searchResultEntry&gt;
                                &lt;/spml:searchResponse&gt;
                        </SPMLSearchRequestResult>
                </SPMLSearchRequestResponse>
        </soap:Body>
</soap:Envelope>
```

Unpacking the Load using XML2 Parser

- The Code :

```
xmlString = system.getScriptText("xmlSearchResult");
//Create an input Stream
is = new java.io.ByteArrayInputStream(xmlString.getBytes("UTF-8"));

//Setup the parser

myParser = system.getParser("ibmdi.XML2");
if (myParser == null) throw "Unable to get ibmdi.XML2";
myParser.setParam("entry.tag",null);
myParser.setParam("value.tag",null);
myParser.setInputStream(is);
myParser.initParser();

myEntry = myParser.readEntry();

mySPMLattr =
myEntry["soap:Envelope"]["soap:Body"]["SPMLSearchRequestResponse"]["SPMLSearchRequestResult"]

task.logmsg(mySPMLattr.getValue());
```

Unpacking the Load using XML2 Parser

- The Output :

```
<spml:searchResponse xmlns:spml="urn:oasis:names:tc:SPML:1:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core" result="urn:oasis:names:tc:SPML:1:0#success">
          <searchResultEntry>
                    <spml:identifier type="URN:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName">
                              <spml:id>QAAF</spml:id>
                    </spml:identifier>
                    <spml:attributes>
                              <dsml:attr name="uid">
                                        <dsml:value>QAAF</dsml:value>
                              </dsml:attr>
                    </spml:attributes>
          </searchResultEntry>
                    ………………
          <searchResultEntry>
                    <spml:identifier type="URN:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName">
                              <spml:id>T9UDK01</spml:id>
                    </spml:identifier>
                    <spml:attributes>
                              <dsml:attr name="uid">
                                        <dsml:value>T9UDK01</dsml:value>
                              </dsml:attr>
                    </spml:attributes>
          </searchResultEntry>
</spml:searchResponse>
```

Processing the Spml using XML2 Parser

```
xmlString = system.getScriptText("xmlSpmlSearchResponse");
//Create an input Stream
is = new java.io.ByteArrayInputStream(xmlString.getBytes("UTF-8"));

//Setup the parser

myParser = system.getParser("ibmdi.XML2");
if (myParser == null) throw "Unable to get ibmdi.XML2";
myParser.setParam("entry.tag",null);
myParser.setParam("value.tag",null);
myParser.setInputStream(is);
myParser.initParser();

myEntry = myParser.readEntry();

for (node in myEntry.getElementsByTagName("dsml:attr")){
           myAttr = node.getAttributes().getNamedItem("name");
           myValue = node.getFirstChild().getValue();
           work.addAttributeValue(myAttr,myValue)
}
```

```
CTGDIS003I *** Start dumping Entry
           Operation: generic
           Entry attributes:
                      uid (replace):        'QAAF'     'QATIMUS'  'QAYFW'     'T9UDK01'
CTGDIS004I *** Finished dumping Entry
```

Security Challenges

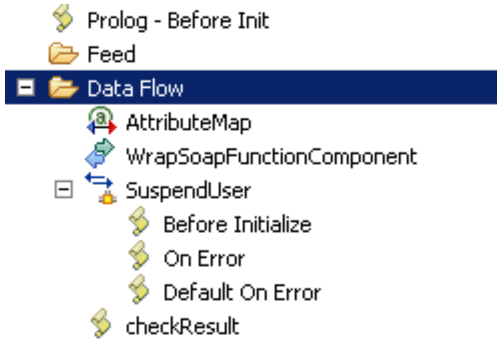
Security is difficult….

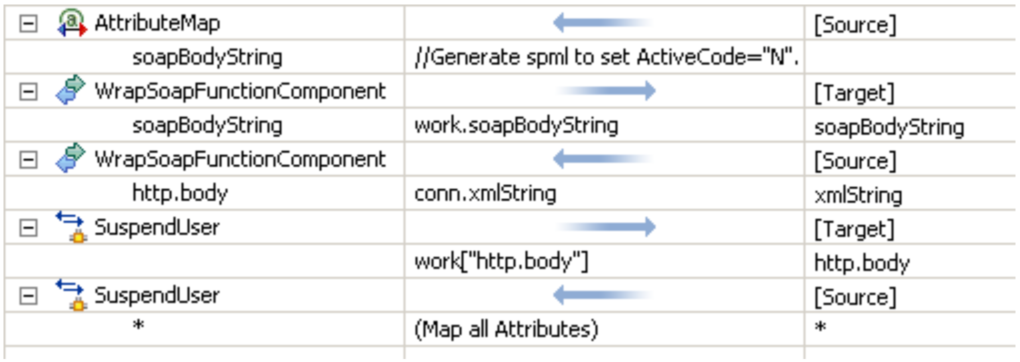
- Many of the Web Service components historically did not support HTTPS
- Current Fixpack level should work, but….
- Remember that not only the WS request/response may require HTTPS, but also WSDL retrieval may use it
- Understand your need for keystores and how to work with certificates
- JMS and SSL is still a new combination
- Axis Components may need additional parameters

Simple Integration Scenario

The Simple Integration Scenario

- ITIM RMI Adapter Suspend Assembly Line
- Setting a fixed value to attribute in SPMLv1
- Wrapping the SPML in a Soap Request
- Wrap the Soap Request in a Soap Body/Envelope
- Call/Reply over HTTPS to Service Provider
- Checking the result


- Prolog - Before Init
- Feed
- Data Flow
  - AttributeMap
  - WrapSoapFunctionComponent
  - SuspendUser
    - Before Initialize
    - On Error
    - Default On Error
  - checkResult



AttributeMap	←	[Source]
soapBodyString	//Generate spml to set ActiveCode="N".	
WrapSoapFunctionComponent	→	[Target]
soapBodyString	work.soapBodyString	soapBodyString
WrapSoapFunctionComponent	←	[Source]
http.body	conn.xmlString	xmlString
SuspendUser	→	[Target]
	work["http.body"]	http.body
SuspendUser	←	[Source]
*	(Map all Attributes)	*

The SPML XML

```
<spml:modifyRequest xmlns:spml="urn:oasis:names:tc:SPML:1:0"
    xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core">

    <spml:identifier type="urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName">

        <spml:id>QAYFW</spml:id>

    </spml:identifier>

    <spml:modifications>

        <dsml:modification name="ActiveCode" operation="replace">

                <dsml:value>N</dsml:value>

        </dsml:modification>

    </spml:modifications>

</spml:modifyRequest>
```

The Soap Request

```
<SPMLModifyRequest xmlns="http://www.kmd.dk/KMD.YH.KSPAabenSpml">
		<request>&lt;spml:modifyRequest xmlns:spml="urn:oasis:names:tc:SPML:1:0" xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"&gt;&#xD;
		&lt;spml:identifier type="urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName"&gt;&#xD;
				&lt;spml:id&gt;QAYFW&lt;/spml:id&gt;&#xD;
		&lt;/spml:identifier&gt;&#xD;
		&lt;spml:modifications&gt;&#xD;
				&lt;dsml:modification name="ActiveCode" operation="replace"&gt;&#xD;
						&lt;dsml:value&gt;N&lt;/dsml:value&gt;&#xD;
				&lt;/dsml:modification&gt;&#xD;
		&lt;/spml:modifications&gt;&#xD;
&lt;/spml:modifyRequest&gt;&#xD;
		</request>
</SPMLModifyRequest>
```

Building The SPML Soap Request

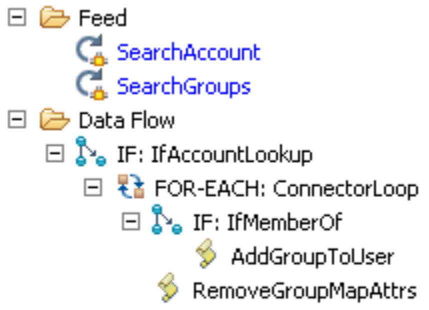
```
//Generate spml to set ActiveCode="N"
spml = system.newEntry();
var modifyRequest = spml.createElementNS("urn:oasis:names:tc:SPML:1:0","spml:modifyRequest");
spml.setAttribute(modifyRequest);
spml["spml:modifyRequest"]["@xmlns:dsml"]="urn:oasis:names:tc:DSML:2:0:core";
spml["spml:modifyRequest"]["spml:identifier"]=null;
spml["spml:modifyRequest"]["spml:identifier"]["@type"]="urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName";
spml["spml:modifyRequest"]["spml:identifier"]["spml:id"]= work.getString("eruid");

spml["spml:modifyRequest"]["spml:modifications"]=null;
spml["spml:modifyRequest"]["spml:modifications"]["dsml:modification"]=null;

spml["spml:modifyRequest"]["spml:modifications"]["dsml:modification"]["@name"]="ActiveCode";
spml["spml:modifyRequest"]["spml:modifications"]["dsml:modification"]["@operation"]="replace";
spml["spml:modifyRequest"]["spml:modifications"]["dsml:modification"]["dsml:value"]="N";

//Wrap the spml in the KMD SPMLModifyRequest
soapBody = system.newEntry();
soapBody["SPMLModifyRequest"] = null;
soapBody["SPMLModifyRequest"]["@xmlns"]="http://www.kmd.dk/KMD.YH.KSPAabenSpml";
soapBody["SPMLModifyRequest"]["request"]=spml.toXML();

ret.value = soapBody.toXML();
```

Complex Integration Scenario
Scripted Web Service Connector

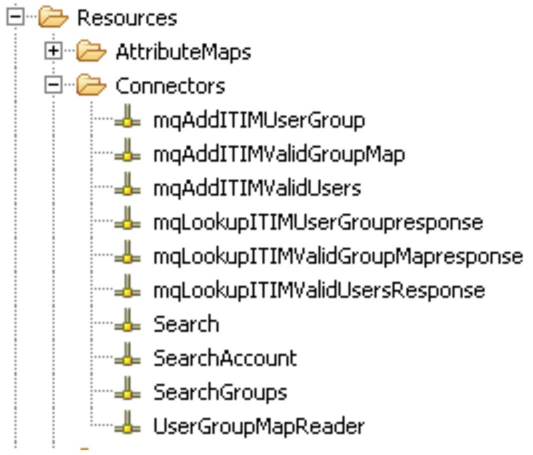
Them Complex Integration Scenario

- ITIM DSML Adapter Search Assembly Line
- User and Group Iterators – Script Connectors
- Account flow includes Connector Loop (scripted) to add group memberships
- Message flow is WebSphere Messge Broker based
  - Soap Request Message posted to JMS (MQ) queue
  - Use the correlation id to look up single response
- Parse the response to map the data

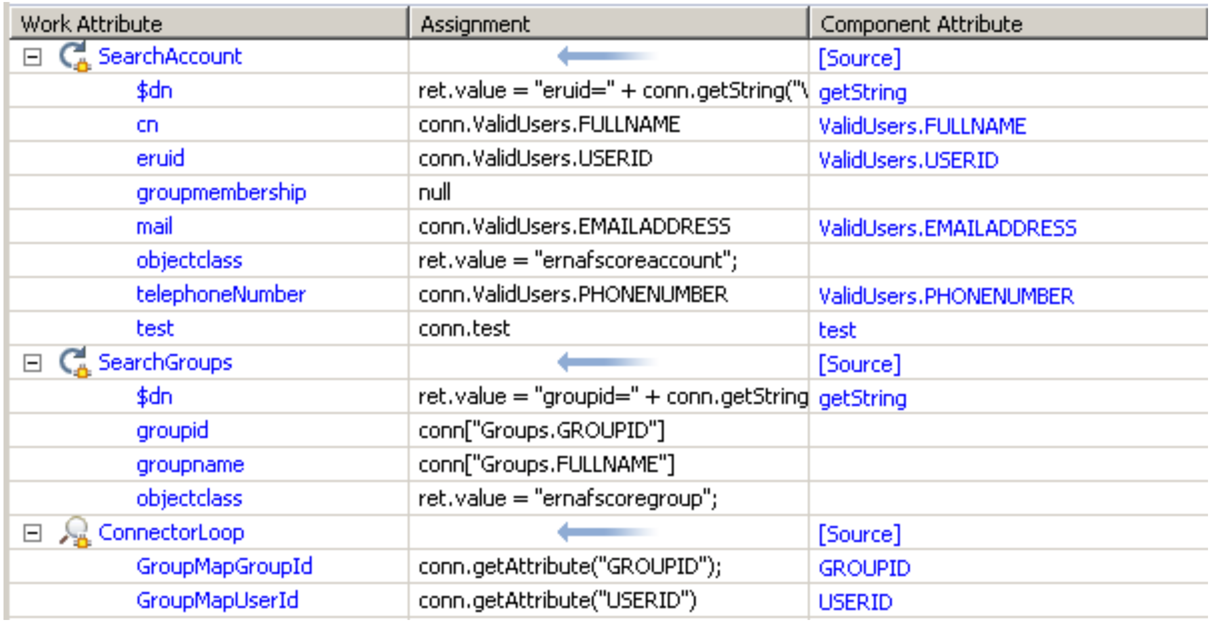
The Complex Scenario - Configuration



```
Feed
    SearchAccount
    SearchGroups
Data Flow
    IF: IfAccountLookup
        FOR-EACH: ConnectorLoop
            IF: IfMemberOf
                AddGroupToUser
            RemoveGroupMapAttrs
```



```
Resources
    AttributeMaps
    Connectors
        mqAddITIMUserGroup
        mqAddITIMValidGroupMap
        mqAddITIMValidUsers
        mqLookupITIMUserGroupresponse
        mqLookupITIMValidGroupMapresponse
        mqLookupITIMValidUsersResponse
        Search
        SearchAccount
        SearchGroups
        UserGroupMapReader
```



Work Attribute	Assignment	Component Attribute
SearchAccount	←	[Source]
$dn	ret.value = "eruid=" + conn.getString("\	getString
cn	conn.ValidUsers.FULLNAME	ValidUsers.FULLNAME
eruid	conn.ValidUsers.USERID	ValidUsers.USERID
groupmembership	null	
mail	conn.ValidUsers.EMAILADDRESS	ValidUsers.EMAILADDRESS
objectclass	ret.value = "ernafscoreaccount";	
telephoneNumber	conn.ValidUsers.PHONENUMBER	ValidUsers.PHONENUMBER
test	conn.test	test
SearchGroups	←	[Source]
$dn	ret.value = "groupid=" + conn.getString	getString
groupid	conn["Groups.GROUPID"]	
groupname	conn["Groups.FULLNAME"]	
objectclass	ret.value = "ernafscoregroup";	
ConnectorLoop	←	[Source]
GroupMapGroupId	conn.getAttribute("GROUPID");	GROUPID
GroupMapUserId	conn.getAttribute("USERID")	USERID

The Complex Scenario – SearchAccount

- Script Connector basics
- Walkthrough of the Script Connector
  - Flow of the functions in the Script Connector
  - Actual Code (for reference)

The Script Connector functions used

- Initialize
  - This function initializes the Connector. It is called before any of the other functions and should contain code that initializes basic parameters, establishes connections, and so forth.
- selectEntries
  - This function is called to prepare the Connector for sequential read. When this function is called it is typically because the Connector is used as an Iterator in an AssemblyLine.
- getNextEntry
  - This function must populate the Entry object with attributes and values from the next entry in the input set. When the Connector has no more entries to return, it must use the result object to signal end-of-input back to the caller.
- findEntry
  - The findEntry function is called to find an entry in the connected system that matches the criteria specified in the search object.

The Complex Scenario – Script initialize flow

- Setup the MQ connectors and associated entry
  - MQ Connector in Add Mode to post request
  - MQ Connector in Lookup Mode to get response
- Setup the Parsers and associated entry
  - Parser to read the User Response
  - Parser to read the Response Result
- Setup status Entry
  - Used to merge the actual user data when parsing the response XML data

The Complex Scenario – Script initialize

```
// Place initialization code before function declarations
//
//Setup the MQ connectors and associated entry
var myMQ_Add = system.getConnector("Connectors/mqAddITIMValidUsers");
if (myMQ_Add == null) throw "Unable to get Connectors/mqAddITIMValidUsers";
var myMQ_Lookup =
system.getConnector("Connectors/mqLookupITIMValidUsersResponse");
if (myMQ_Lookup == null) throw "Unable to get
Connectors/mqLookupITIMValidUsersResponse";
var myEntry = system.newEntry();
//
//Setup the Parsers and associated entry
//
var myParser = system.getParser("Parsers/ITIMValidUsers");
if (myParser == null) throw "Unable to get Parsers/ITIMValidUsers";
var myResponseParser = system.getParser("Parsers/responseData");
if (myResponseParser == null) throw "Unable to get Parsers/responseData";

var myITIMValidUsersEntry = "";

//Setup status Entry
var statusEntry = system.newEntry();
```

The Complex Scenario – selectEntries()

```
function selectEntries()
{
            task.logmsg("DEBUG","selectEntries started");

            //Setup the MQ Add queue and send ITIMValidUsers XML
            myMQ_Add.initialize(null);
            var myAddEntry = system.newEntry();
            var myITIMValidUsers = system.getScriptText("ITIMValidUsers");
            var myDate = javax.xml.bind.DatatypeConverter.printDateTime(java.util.Calendar.getInstance());
            //Replace control data
            myITIMValidUsers = myITIMValidUsers.replaceAll(">control:ProcessID<",">ITIM<");
            myITIMValidUsers = myITIMValidUsers.replaceAll(">control:EnterpriseUserID<",">ITIMUSER<");
            myITIMValidUsers = myITIMValidUsers.replaceAll(">control:InitiatingComponent<",">ITIMADAPTER<");
            myITIMValidUsers = myITIMValidUsers.replaceAll(">2001-12-31T12:00:00\\+02:00<",">" + myDate + "<");

            myAddEntry.setAttribute("message",myITIMValidUsers);
            myMQ_Add.putEntry(myAddEntry);
            myMessageid = myAddEntry.getProperty("$jms.messageid");
            task.logmsg("DEBUG","Message ID : " + myAddEntry.getProperty("$jms.messageid"));

            //Setup the response queue and pull out the response
            myMQ_Lookup.initialize(null);
            mySearchCriteria = new
com.ibm.di.server.SearchCriteria("jms.JMSCorrelationID",com.ibm.di.server.SearchCriteria.EXACT,myMessageid);
            task.logmsg("DEBUG","Found : " + myMQ_Lookup.getFindEntryCount());
            myEntry = myMQ_Lookup.findEntry(mySearchCriteria);
            task.logmsg("DEBUG","Message returned : " + myEntry.getString("message"));
```

The Complex Scenario – selectEntries() cont.

```
//Setup the data
//
//This is the real handling of the response
//remember to add a newline as the transform may fail without it
responseXML = myEntry.getString("message") + "\n";
responseStatus = system.xslTransform(system.getScriptText("xslResponseStatus"),responseXML);

//Create an input Stream
is = new java.io.ByteArrayInputStream(responseStatus.getBytes("UTF-8"));

//Setup the SAX parser
myResponseParser.setInputStream(is);
myResponseParser.initParser();

var myResponseStatusEntry = myResponseParser.readEntry();
while (myResponseStatusEntry != null){
           statusEntry.merge(myResponseStatusEntry);
           myResponseStatusEntry = myResponseParser.readEntry()
}
task.logmsg("DEBUG","Status : " + statusEntry);
```

The Complex Scenario – selectEntries() cont.

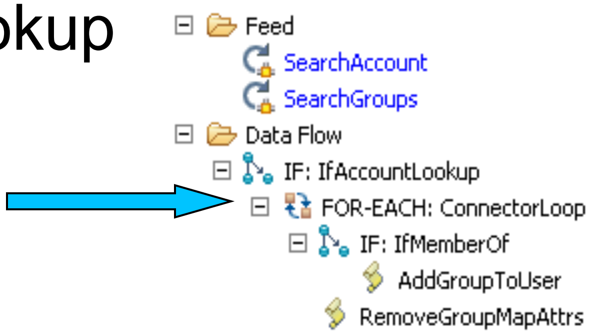
```
           if (!(statusEntry.getString("Status").equalsIgnoreCase("ok"))){
                      task.logmsg("Status : " + statusEntry.getString("Status"));
                      //throw "Status is not OK";
                      result.setStatus(2);
                      result.setMessage("Status is not OK");
                      break;
           } else {

                      //Setup the for the actual responseData

                      ITIMValidUsersResponseData =
system.xslTransform(system.getScriptText("xslITIMValidUsersResponseData"),responseXML);
                      //Namespaced prefix not used anyhow...
                      //ITIMValidUsersResponseData =
system.xslTransform(system.getScriptText("xslRemoveNameSpaces"),ITIMValidUsersResponseData);

                      //Create an input Stream
                      is1 = new java.io.ByteArrayInputStream(ITIMValidUsersResponseData.getBytes("UTF-8"));

                      //Setup the parser
                      myParser.setInputStream(is1);
                      myParser.initParser();

                      myITIMValidUsersEntry = myParser.readEntry();
           }
}
```

The Complex Scenario – getNextEntry()

```
function getNextEntry ()
{
		if (myITIMValidUsersEntry != null){
			//system.dumpEntry(myITIMValidUsersEntry);
			entry.merge(myITIMValidUsersEntry);
			myITIMValidUsersEntry = myParser.readEntry();
		} else {
			result.setStatus (0);
			result.setMessage ("End of input");
		}
}
```

The Complex Scenario – findEntry()

- This is from the Connector Loop Lookup Mode Connector called
- Only the findEntry() shown
- Flow :
  - Get the supplied Search Criteria
  - Run the AssemblyLine that creates the Soap request
  - Setup the MQ Add queue and post XML
  - Setup the response queue and pull out the response
  - Transform the response and setup the parser
  - Process the response



The Complex Scenario – findEntry()

```
function findEntry ()
{
          task.logmsg("DEBUG","findEntry started");

          var criteria = search.getCriteria();
          var findUserID = criteria.get(0).value; //only use the first search criteria
          task.logmsg("DEBUG","Userid to find groups for : " + findUserID)

          connector.clearFindEntries();

          //Run the ITIMValidGroupMap Al to create the relevant message for the Queue
          var myUserEntry = system.newEntry();
          myUserEntry.setAttribute("myUserId",findUserID);

          myAl = main.startAL("ITIMValidGroupMap",myUserEntry);
          myAl.join();
          myRecEntry = myAl.getCurrentWork();
          task.logmsg("DEBUG","ITIMValidGroupMap xml : " + myRecEntry.getString("myXML"));
```

The Complex Scenario – findEntry() cont.

```
          //Setup the MQ Add queue and send ITIMUserGroup XML
          myMQ_Add.initialize(null);
          var myAddEntry = system.newEntry();

          myAddEntry.setAttribute("message",myRecEntry.getString("myXML"));
          myMQ_Add.putEntry(myAddEntry);
          myMessageid = myAddEntry.getProperty("$jms.messageid");
          task.logmsg("DEBUG","Message ID : " + myAddEntry.getProperty("$jms.messageid"));

          //Setup the response queue and pull out the response
          myMQ_Lookup.initialize(null);
          mySearchCriteria = new 
com.ibm.di.server.SearchCriteria("jms.JMSCorrelationID",com.ibm.di.server.SearchCriteria.EXACT,myMessag
eid);
          task.logmsg("DEBUG","Found : " + myMQ_Lookup.getFindEntryCount());
          myEntry = myMQ_Lookup.findEntry(mySearchCriteria);
          task.logmsg("DEBUG","Message returned : " + myEntry.getString("message"));
```

The Complex Scenario – findEntry() cont.

```
//Setup the data
//
//This is the real handling af the respons
responseXML = myEntry.getString("message") + "\n";
responseStatus = system.xslTransform(system.getScriptText("xslResponseStatus"),responseXML);

//Create an input Stream
is = new java.io.ByteArrayInputStream(responseStatus.getBytes("UTF-8"));

//Setup the parser
myResponseParser.setInputStream(is);
myResponseParser.initParser();

var myResponseStatusEntry = myResponseParser.readEntry();
while (myResponseStatusEntry != null){
           statusEntry.merge(myResponseStatusEntry);
           myResponseStatusEntry = myResponseParser.readEntry()
}
```

The Complex Scenario – findEntry() cont.

```
    if (!(statusEntry.getString("Status").equalsIgnoreCase("ok"))){
            task.logmsg("DEBUG","Status : " + statusEntry.getString("Status"));
            //throw "Status is not OK";
            result.setStatus(2);
            result.setMessage("Status is not OK");
            break;
    } else {

            //Setup the for the actual responseData

            ITIMValidGroupMapresponseData = system.xslTransform(system.getScriptText("xslITIMValidGroupMapresponseData"),responseXML);
            ITIMValidGroupMapresponseData = system.xslTransform(system.getScriptText("xslRemoveNameSpaces"),ITIMValidGroupMapresponseData);
            //Create an input Stream
            is1 = new java.io.ByteArrayInputStream(ITIMValidGroupMapresponseData.getBytes("UTF-8"));

            //Setup the parser
            myParser.setInputStream(is1);
            myParser.initParser();

            myITIMValidGroupMapResponseEntry = myParser.readEntry();
    }

    task.logmsg("DEBUG","myITIMValidGroupMapResponseEntry : " + myITIMValidGroupMapResponseEntry);
    connector.clearFindEntries();
    while (myITIMValidGroupMapResponseEntry != null){
            myITIMValidGroupMapResponseEntry.setAttributeValues("USERID",findUserID);
            task.logmsg("DEBUG","LOOP - myITIMValidGroupMapResponseEntry : " + myITIMValidGroupMapResponseEntry);
            task.logmsg("DEBUG","LOOP - entry : " + entry);
            connector.addFindEntry(myITIMValidGroupMapResponseEntry)
            entry.merge(myITIMValidGroupMapResponseEntry);
            myITIMValidGroupMapResponseEntry = myParser.readEntry();
    }

    if (connector.getFindEntryCount() == 1)
            result.setStatus(1);
    else
            result.setStatus(0);

}
```

Other Reference Material

Material from TDI Users

- TDI Users Website has a couple of HowTos :
  - http://www.tdi-users.org/twiki/bin/view/Integrator/HowTo
    - **(Very) Advanced XML Handling – from ETTC 2010**
    - **XML and WebService lecture by Lak Sri**
    - **WebService presention from IBM Support L2**

Q & A