WORKING WITH WEB SERVICES

Rob Richards

http://xri.net/=rob.richards

www.cdatazone.org

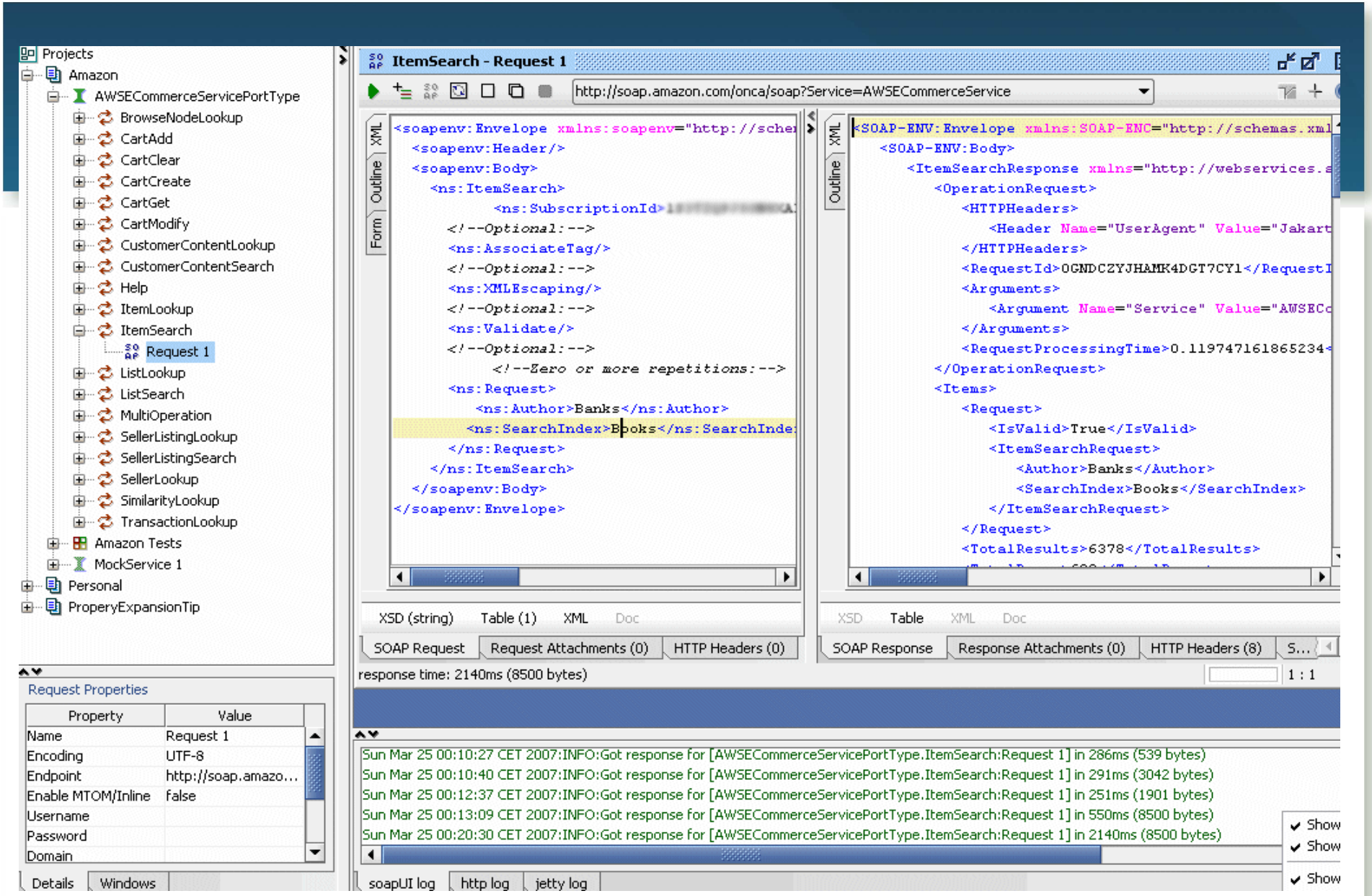
ZEND/PHP CONFERENCE & EXPO 8-11 OCTOBER 2007

Helpful Tools

- soapUI
  - http://www.soapui.org/
  - Multiple platforms
  - Free & enhanced Pro versions
- SOAPSonar
  - http://www.crosschecknet.com/
  - Windows only
  - Personal (Free) and Professional versions
- Misc. XML Editors
  - XML NotePad 2007 (Microsoft)
  - XML Spy (Altova)

soapUI Features

- Web Service Inspection
  - WSDL Inspector
  - HTTP data inspection (request & response)
- Web Service Invocation
  - Automatic Request generation
  - Authentication support (Basic, Digest, WS-Security)
  - Custom HTTP Header support
- Web Service Development and Validation
- Web Service Functional Testing
- Web Service Load Testing
- Web Service Simulation


Projects
Amazon
AWSECommerceServicePortType
BrowseNodeLookup
CartAdd
CartClear
CartCreate
CartGet
CartModify
CustomerContentLookup
CustomerContentSearch
Help
ItemLookup
ItemSearch
Request 1
ListLookup
ListSearch
MultiOperation
SellerListingLookup
SellerListingSearch
SellerLookup
SimilarityLookup
TransactionLookup
Amazon Tests
MockService 1
Personal
ProperyExpansionTip
ItemSearch - Request 1
http://soap.amazon.com/onca/soap?Service=AWSECommerceService
XML
Outline
Form
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header/>
<soapenv:Body>
<ns:ItemSearch>
<ns:SubscriptionId>
<!--Optional:-->
<ns:AssociateTag/>
<!--Optional:-->
<ns:XMLEscaping/>
<!--Optional:-->
<ns:Validate/>
<!--Optional:-->
<!--Zero or more repetitions:-->
<ns:Request>
<ns:Author>Banks</ns:Author>
<ns:SearchIndex>Books</ns:SearchIndex>
</ns:Request>
</ns:ItemSearch>
</soapenv:Body>
</soapenv:Envelope>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<ItemSearchResponse xmlns="http://webservices.amazon.com/AWSECommerceService/2007-03-26">
<OperationRequest>
<HTTPHeaders>
<Header Name="UserAgent" Value="Jakarta Commons-HttpClient/3.0"/>
</HTTPHeaders>
<RequestId>0GNDCZYJHAMK4DGT7CY1</RequestId>
<Arguments>
<Argument Name="Service" Value="AWSECommerceService"/>
</Arguments>
<RequestProcessingTime>0.119747161865234</RequestProcessingTime>
</OperationRequest>
<Items>
<Request>
<IsValid>True</IsValid>
<ItemSearchRequest>
<Author>Banks</Author>
<SearchIndex>Books</SearchIndex>
</ItemSearchRequest>
</Request>
<TotalResults>6378</TotalResults>
XSD (string) Table (1) XML Doc
SOAP Request Request Attachments (0) HTTP Headers (0)
XSD Table XML Doc
SOAP Response Response Attachments (0) HTTP Headers (8) S...
response time: 2140ms (8500 bytes)
1 : 1
Request Properties
Property Value
Name Request 1
Encoding UTF-8
Endpoint http://soap.amazo...
Enable MTOM/Inline false
Username
Password
Domain
Details Windows
Sun Mar 25 00:10:27 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 286ms (539 bytes)
Sun Mar 25 00:10:40 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 291ms (3042 bytes)
Sun Mar 25 00:12:37 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 251ms (1901 bytes)
Sun Mar 25 00:13:09 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 550ms (8500 bytes)
Sun Mar 25 00:20:30 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 2140ms (8500 bytes)
Show
Show
Show
soapUI log http log jetty log

Making SOAP Requests

Tips and Tricks working with the SoapClient

Canadian Cattle Identification Agency (CCIA)

- Handles cattle age verification
- Provides a centralized database where producers can maintain their cattle information
- Information is readily available to help speed up import/exports
- Information can be correlated with RFID tags
- System is entirely online
- SOAP is used for application integration
- Website: http://www.canadaid.ca/

CCIA: Retrieve API Information

```
$wsdl = 'https://www.clia.livestockid.ca/CLTS/public/help/en/
webservices/WebServiceAnimalSearch/IAnimalSearchWSv2.wsdl';

@$client = new SOAPClient($wsdl);

$types = $client->__getTypes();
    foreach ($types AS $type) {
    print $type."\n\n";
}

$functions = $client->__getFunctions();
    foreach ($functions AS $function) {
    print $function."\n\n";
}
```

CCIA: Can We Be Any More Descriptive?

```
com_clarkston_cts_webservice_animal_search_value_AnimalSearchResultWSValue
    executeSearch
    (com_clarkston_cts_webservice_animal_search_value_AnimalSearchCriteriaWSV
    alue $criteria)

struct
    com_clarkston_cts_webservice_animal_search_value_AnimalSearchCriteriaWSV
    alue {
 ArrayOfcom_clarkston_cts_webservice_value_TagRangeWSValue tagRangeList;
 string publicAccountId;
}

struct com_clarkston_cts_webservice_value_TagRangeWSValue {
 string tagType;
 string endTag;
 string startTag;
}

com_clarkston_cts_webservice_value_TagRangeWSValue
    ArrayOfcom_clarkston_cts_webservice_value_TagRangeWSValue[]
```

CCIA: Functions and Types Translated

```
AnimalResult executeSearch(AnimalCriteria $criteria)

struct TagRange {
    string tagType;
    string endTag;
    string startTag;
}

TagRange ArrayOfTagRange[]

struct AnimalCriteria {
    ArrayOfTagRange    tagRangeList;
    string             publicAccountId;
}
```

CCIA: Making The Request

```
try {
    $client = new SoapClient($wsdl);
    $stag = '000000124000050000102';
    $etag = '000000124000050000106';

    $tag1 = array('startTag'=>$stag, 'endTag'=>$etag, 'tagType'=>'C');
    $tRangeList = array($tag1);

    $res = $client->executeSearch(array('tagRangeList'=>$tRangeList,
                                        'publicAccountId'=>""));
    var_dump($res);
} catch (SoapFault $e) {
    var_dump($e);
}
```

CCIA: DOH! . . . Denied

```
object(SoapFault)#3 (9) {
  . . .
  ["faultstring"]=> string(273) "No Deserializer found to
    deserialize a
    'http://com.clarkston.cts.webservice.animal.search/IAni
    malSearchWSv2.xsd:ArrayOfcom_clarkston_cts_webs
    ervice_value_TagRangeWSValue' using encoding
    style 'http://schemas.xmlsoap.org/soap/encoding/'.
    [java.lang.IllegalArgumentException]"

  ["faultcode"]=> string(15) "SOAP-ENV:Client"
  ["faultactor"]=> string(22) "/CLTS/AnimalSearchWSv2"
}
```

CCIA: Capturing the Request

```
$client_options = array ('trace'=>1);
$client = new SoapClient($wsdl, $client_options);
try {
    /* code edited for brevity */

    $res = $client->executeSearch(array('tagRangeList'=>$tRangeList,
                                        'publicAccountId'=>""));

} catch (SoapFault $e) {
    print $client->__getLastRequest();
}
```

CCIA: Capturing Messages

Be careful when capturing the output from __getLastRequest() or __getLastResponse()

- Pipe the output to a file

  php myclient.php > request.php

- Capture it to a variable

```
$request = $client->__getLastRequest();
file_put_contents('request.xml', $request);
```

CCIA: The SOAP Request (edited)

```
<ns1:executeSearch>
  <criteria xsi:type="ns2:AnimalSearchCriteriaWSValue">
    <tagRangeList SOAP-ENC:arrayType="ns2:TagRange[1]"
                          xsi:type="ns2:ArrayOfTagRange">
      <item xsi:type="ns2:TagRange">
        <tagType xsi:type="xsd:string">C</tagType>
        <endTag xsi:type="xsd:string">. . .000106</endTag>
        <startTag xsi:type="xsd:string">. . .000102</startTag>
      </item>
    </tagRangeList>
    <publicAccountId xsi:type="xsd:string"/>
  </criteria>
</ns1:executeSearch>
```

CCIA: Accessing The Request

```
class CustomSoapClient  extends SoapClient {
   function __doRequest($request, $location,
                                $saction, $version) {
      $doc = new DOMDocument();
      $doc->loadXML($request);

      /* Modify SOAP Request held within $doc */

      return parent::__doRequest($doc->saveXML(),
                                   $location, $saction, $version);
   }
}
```

CCIA: Modifying The Request

```
$root = $doc->documentElement;
$encprefix = $root->lookupPrefix('http://schemas.xmlsoap.org/soap/encoding/');
$xpath = new DOMXPath($doc);
$xpath->registerNamespace("myschem", 'http://. . ./XMLSchema-instance');
$xpath->registerNamespace("myxsi", 'http://schemas.xmlsoap.org/soap/. . .');

$query = '/*/*/*//*[@myxsi:arrayType and @myschem:type]';

if ($nodelists = $xpath->query($query)) {
   foreach ($nodelists AS $node) {
      if ($attnode =
        $node->getAttributeNodeNS('http://www.w3.org/2001/XMLSchema-instance',
                                                  'type'))
      {
              $attnode->nodeValue = $encprefix.':Array';
```

CCIA: The Altered Request

```
<ns1:executeSearch>
  <criteria xsi:type="ns2:AnimalSearchCriteriaWSValue">
    <tagRangeList SOAP-ENC:arrayType="ns2:TagRange[1]"
                  xsi:type="SOAP-ENC:Array">
      <item xsi:type="ns2:TagRange">
        <tagType xsi:type="xsd:string">C</tagType>
        <endTag xsi:type="xsd:string">. . .000106</endTag>
        <startTag xsi:type="xsd:string">. . .000102</startTag>
      </item>
    </tagRangeList>
    <publicAccountId xsi:type="xsd:string"/>
  </criteria>
</ns1:executeSearch>
```

CCIA: How Did You Get The Request?

```
$client_options = array ('trace'=>1);
$client = new CustomSoapClient($wsdl, $client_options);
try {
    /* code edited for brevity */

    $res = $client->executeSearch(array('tagRangeList'=>$tRangeList,
                                        'publicAccountId'=>""));
    $res->__getLastRequest() /* Still shows bad message */
```

- Messages altered within __doRequest() are not reflected when __getLastRequest() is called
- View the raw XML from the DOMDocument within __doRequest

```
print $doc->saveXML();
```

CCIA: Handling SOAP-ENC:Array w/ PHP 5.2

```
$client_options = array(
                    'features'=>SOAP_USE_XSI_ARRAY_TYPE);

$client = new SoapClient($wsdl, $client_options);
$stag = '0000001240000050000102';
$etag = '0000001240000050000106';

$tag1 = array('startTag'=>$stag, 'endTag'=>$etag,
                                        'tagType'=>'C');
$tRangeList = array($tag1);

$res = $client->executeSearch(array('tagRangeList'=>$tRangeList,
                                'publicAccountId'=>""));
```

ExactTarget

- Provide tools and services for email communications
- Focus on Email Marketing
- On-Demand Access
- Personal Interaction via Web-based GUI
- Application integration via REST and/or SOAP
- SOA based SOAP interface

ExactTarget: Retrieve API information

```
$wsdl = 'https://webservice.exacttarget.com/etframework.wsdl';

@$client = new SOAPClient($wsdl, array('trace'=>1));

$types = $client->__getTypes();
foreach ($types AS $type) {
        print $type."\n\n";
}

$functions = $client->__getFunctions();
foreach ($functions AS $function) {
        print $function."\n\n";
}
```

ExactTarget Functions

CreateResponse Create(CreateRequest $parameters)

RetrieveResponseMsg Retrieve(RetrieveRequestMsg $parameters)

UpdateResponse Update(UpdateRequest $parameters)

DeleteResponse Delete(DeleteRequest $parameters)

ExecuteResponseMsg Execute(ExecuteRequestMsg $parameters)

ExactTarget: RetrieveRequestMsg

```
struct RetrieveRequestMsg {
    RetrieveRequest RetrieveRequest;
}

struct RetrieveRequest {
    ClientID ClientIDs;
    string ObjectType;
    string Properties;
    FilterPart Filter;
    AsyncResponse RespondTo;
    APIProperty PartnerProperties;
    string ContinueRequest;
    boolean QueryAllAccounts;
    boolean RetrieveAllSinceLastBatch;
}
```

ExactTarget: Basic List Retrieval Request

```
@$client = new ExactTargetSoapClient($wsdl, $options);
/* WS-Security Username handling */

$request = new ExactTarget_RetrieveRequestMsg();

$rr = new ExactTarget_RetrieveRequest();
$rr->ObjectType = "List";
$rr->Properties = array("ListName");

$request->RetrieveRequest = $rr;

$response = $client->Retrieve($request);
```

ExactTarget: List Retrieval Response

```
object(stdClass)#6 (3) {
  ["OverallStatus"]=> string(2) "OK"
  ["RequestID"]=> string(36) "6bb27c71-16e6-4167-a57c-32df045174c4"
  ["Results"]=>
  array(2) {
    [0]=> object(stdClass)#12 (3) {
      ["PartnerKey"]=> NULL
      ["ObjectID"]=> NULL
      ["ListName"]=> string(8) "RobsList"
    }
    [1]=> object(stdClass)#8 (3) {
      ["PartnerKey"]=> NULL
      ["ObjectID"]=> NULL
      ["ListName"]=> string(12) "New Rob List"
```

ExactTarget: FilterPart Definition

From RetrieveRequest Definition:

FilterPart Filter;

FilterPart Definition:

struct FilterPart { }

ExactTarget: Using a FilterPart

```
<complexType name="SimpleFilterPart">
  <complexContent>
    <extension base="tns:FilterPart">
      <sequence>
        <element name="Property" type="xsd:string"
                 minOccurs="1" maxOccurs="1" />
        <element name="SimpleOperator" type="tns:SimpleOperators"
                 minOccurs="1" maxOccurs="1" />
        <element name="Value" type="xsd:string"
                 minOccurs="0" maxOccurs="unbounded" />
        <element name="DateValue" type="xsd:dateTime"
                 minOccurs="0" maxOccurs="unbounded" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

ExactTarget: Filtering Requests

```
$sfp = new ExactTarget_SimpleFilterPart();
$sfp->Property = "ListName";
$sfp->SimpleOperator =
    ExactTarget_SimpleOperators::equals;
$sfp->Value = array("RobsList");


$rr->Filter = $sfp;
```

ExactTarget: Filter Results

```
object(stdClass)#7 (2) {
  ["OverallStatus"]=>
  string(60) "Error: Object reference not set to an instance
    of an object."
  ["RequestID"]=>
  string(36) "9c7ebf66-d211-43cf-9984-0adfb4b1e476"
}
```

This Doesn't Look Right!

ExactTarget: Examining The Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=". . ."
        xmlns:ns1="http://exacttarget.com/wsdl/partnerAPI">
<SOAP-ENV:Body>

  <ns1:RetrieveRequestMsg>
    <ns1:RetrieveRequest>
      <ns1:ObjectType>List</ns1:ObjectType>
      <ns1:Properties>ListName</ns1:Properties>

      <ns1:Filter/> <!-- Where is our Filter??? -->

    </ns1:RetrieveRequest>
  </ns1:RetrieveRequestMsg>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ExactTarget: Using The SoapVar Class

```
class SoapVar {
   __construct ( mixed $data,
                 int $encoding
                 [, string $type_name
                 [, string $type_namespace
                 [, string $node_name
                 [, string $node_namespace]]]] )
}
```

ExactTarget: Using The SoapVar Class

```
$soapvar = new SoapVar(
    $sfp, /* data */
    SOAP_ENC_OBJECT, /* encoding */
    'SimpleFilterPart', /* type_name */
   "http://exacttarget.com/wsdl/partnerAPI" /* type ns */
);

$rr->Filter = $soapvar;
```

ExactTarget: Filtered Results

```
object(stdClass)#8 (3) {
  ["OverallStatus"]=>string(2) "OK"
  ["RequestID"]=>string(36) "638f2307-xxxxxxxx"
  ["Results"]=>
  object(stdClass)#14 (3) {
    ["PartnerKey"]=>NULL
    ["ObjectID"]=>NULL
    ["ListName"]=>string(8) "RobsList"
  }
}
```

ExactTarget: Examining The Request (Take 2)

```
<!-- Only internal struct of the envelope displayed -->
<ns1:RetrieveRequest>
   <ns1:ObjectType>List</ns1:ObjectType>
   <ns1:Properties>ListName</ns1:Properties>

   <ns1:Filter xsi:type="ns1:SimpleFilterPart">
      <ns1:Property>ListName</ns1:Property>
      <ns1:SimpleOperator>equals</ns1:SimpleOperator>
      <ns1:Value>RobsList</ns1:Value>
   </ns1:Filter>

</ns1:RetrieveRequest>
```

Server Sided SOAP

Working through server sided issues

SoapServer: WSDL Caching

DISABLE WSDL CACHING WHEN DEVELOPING!

```
ini_set("soap.wsdl_cache_enabled", "0");
```

SoapServer: Handling Requests

- The typical method to process a SOAP request
  $soapServer->handle();
- Request may be passed to handler
  $soapServer->handle($request);
- Passing in the request is handy in a number of cases
  - You can be guaranteed of the same request while debugging
  - Debugging can be performed via CLI
  - Requests can be modified prior to being processed

SoapServer: Handling Requests

```
/* create the initial request to be re-used */
$request = file_get_contents("php://input");
file_save_contents('debugging.xml', $request);

/* retrieve the saved request on subsequent calls
$request = file_get_contents('debugging.xml');
*/

$server = new SoapServer($wsdl);
/* Setup function handlers */
$server->handle($request);
```

Swiss Army SOAP

When All Else Fails

Typemap: Custom Serial/De-Serialization

- A SOAP option for both client and server
- Specifies functions to use to convert objects to XML (to_xml) and XML to objects (from_xml) based on type

```
$typemap = array(
   array("type_name" => <name of type>,
         "type_ns"     => <namespace of type>,
         "to_xml"      => <function to convert obj to xml>,
         "from_xml"    => <function to convert xml to obj>
   )
);
```

Typemap: WSDL function and Types

```
struct Quote {
    string Symbol;
    float Quote;
}

struct Quotes {
    Quote Quote;
}

struct Companies {
    string Symbol;
}

Quotes getQuote(Companies $parameters)
```

Typemap: SoapClient Example

```
function compserial($obj) {
    return '<Companies xmlns="http://example.org/StockQuote">
                <Symbol>msft</Symbol>
            </Companies>';
}

$typemap = array(
   array("type_name" => "Companies",
         "type_ns"   => "http://example.org/StockQuote",
         "to_xml"    => "compserial"));

$options = array("typemap"=>$typemap);
$client = new SoapClient('localtest.wsdl', $options);

$test = $client->getQuote(array("Symbol" => array("ibm")));
```

Typemap: SoapClient Request

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="http://example.org/StockQuote">
   <SOAP-ENV:Body>
      <ns1:Companies xmlns="http://example.org/StockQuote">
         <Symbol>msft</Symbol>
      </ns1:Companies>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Typemap: SoapServer Example

```
function quoteToXML($obj) {
   return '<Quotes xmlns="http://example.org/StockQuote">'.
        '<Quote><Symbol>yhoo</Symbol><Quote>35</Quote></Quote>'.
        '</Quotes>';
}

function getQuote($objs) {
    return array("Quote"=>array(array('Symbol'=>'ibm', 'Quote'=>1)));
}

$tmap = array(array("type_name"=>"Quotes",
                    "type_ns"=>"http://example.org/StockQuote",
                    "to_xml"=>"quoteToXML"));

$server = new SoapServer('localtest.wsdl', array("typemap"=>$tmap));
$server->addFunction("getQuote");
$server->handle();
```

Typemap: SoapServer Response

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="http://example.org/StockQuote">
   <SOAP-ENV:Body>
      <ns1:Quotes xmlns="http://example.org/StockQuote">
         <Quote>
            <Symbol>yhoo</Symbol>
            <Quote>35</Quote>
         </Quote>
      </ns1:Quotes>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Questions?

WORKING
WITH
WEB SERVICES

Rob Richards
http://xri.net/=rob.richards
http://www.cdatazone.org