

Web Services-Working With Temporary GlobalIDs

This documentation was written by Andrew Lindsey (<http://www.lindesay.co.nz>) in 2006 as part of supported code in the LEWOSTuff framework, but this material has been transcribed here. It was written around the time of WebObjects 5.2 and 5.3 on the 1.4 JVM.

GlobalID's are unique identifiers for EO's in the EOF environment. The WebServices framework is able to serialise and deserialise instance of EOKeyGlobalID, but not EOTemporaryGlobalID. This presents a considerable problem working from the client on a remote EO graph.

Problem Description

I am building a graph of inserted/modified/referenced EO-s in the session's default EC and then persisting the changes. In order to reference inserted objects in the EC to create relationships between in-memory EO-s, I am referring to them using their temporary EOGlobalID's. For example...

```
boolean addFoo(  
    String wosid,  
    String name,  
    String code,  
    EOGlobalID barGlobalID)
```

...where 'barGID' may be a temporary GID (EOTemporaryGlobalID) or may be a key GID (EOKeyGlobalID). The temporary GID's are serialising out to the client from the WO application without any difficulty. In this case, the XML chunk in the SOAP envelope looks like this...

```
<barGlobalID xsi:type="ns4:EOGlobalID">  
  <data  
xsi:type="xsd:base64Binary">AAB/AAABAAAXcQEAAAABC8pbeE6Acu4F</data>  
  </barGlobalID>
```

However when I try and use this temporary GID by sending it from the client, back into the WO application for deserialisation, I end up sending this...

```
<barGlobalID xsi:type="SOAP-ENC:Dictionary">  
  <data xsi:type="xsd:base64Binary">  
AAB/AAABAAAXcQEAAAABC8pbeE6Acu4F</data>  
  </barGlobalID>
```

[WO:I am using Apple's WebServicesCore framework and I'm not entirely sure why it is putting whitespace into the base64 encoding, but I have done independent tests which would tend to indicate that this is deserialised correctly by the deserialiser on the WO-AXIS end.]

Passing in 'null' for this argument results in no exception, passing in a temporary GID 'dictionary structure' as shown above produced the following exception and corresponding AXIS fault as shown below...

```
- IllegalArgumentException:
java.lang.IllegalArgumentException
  at
sun.reflect.UnsafeObjectFieldAccessorImpl.set(UnsafeObjectFieldAccessorImp
l.java:63)
  at java.lang.reflect.Field.set(Field.java:519)
  at org.apache.axis.encoding.FieldTarget.set(FieldTarget.java:91)
  at
org.apache.axis.encoding.DeserializerImpl.valueComplete(DeserializerImpl.j
ava:282)
  at
org.apache.axis.encoding.DeserializerImpl.endElement(DeserializerImpl.java
:541)
  at
org.apache.axis.encoding.DeserializationContextImpl.endElement(Deserializa
tionContextImpl.java:1015)
  at
org.apache.axis.message.SAX2EventRecorder.replay(SAX2EventRecorder.java:20
4)
  at
org.apache.axis.message.MessageElement.publishToHandler(MessageElement.jav
a:722)
  at org.apache.axis.message.RPCElement.deserialize(RPCElement.java:233)
  at org.apache.axis.message.RPCElement.getParams(RPCElement.java:347)
  at
org.apache.axis.providers.java.RPCProvider.processMessage(RPCProvider.java
:184)
  at
org.apache.axis.providers.java.JavaProvider.invoke(JavaProvider.java:333)
  at
org.apache.axis.strategies.InvocationStrategy.visit(InvocationStrategy.jav
a:71)
  at org.apache.axis.SimpleChain.doVisiting(SimpleChain.java:150)
  at org.apache.axis.SimpleChain.invoke(SimpleChain.java:120)
  at org.apache.axis.handlers.soap.SOAPService.invoke(SOAPService.java:481)
  at org.apache.axis.server.AxisServer.invoke(AxisServer.java:323)
  at
com.webobjects.appserver._private.WOWebService.performActionNamed(WOWebSer
vice.java:375)
```

If I send a key GID as follows...

```

<barGlobalID xsi:type="SOAP-ENC:Dictionary">
  <primaryKeys SOAP-ENC:arrayType="xsd:anyType[1]"
xsi:type="SOAP-ENC:Array">
    <item_0 xsi:type="xsd:int">1</item_0>
  </primaryKeys>
  <entityName xsi:type="xsd:string">BarEntity</entityName>
</barGlobalID>

```

...then it deserialises without issue. When I try to get a WSDL from a registered web service that has EOGlobalID's as parameters, I tend to get the following exception and corresponding AXIS fault. For this reason it is not possible to find out how the EOGlobalID should be communicated.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
  <soapenv:Fault>
    <faultcode>soapenv:Server.userException</faultcode>
    <faultstring>WSDLException: faultCode=OTHER_ERROR: Can't find
prefix for 'http://www.apple.com/webobjects/webservices/soap/'.
Namespace prefixes must be set on the Definition object using the
addNamespace(...) method.: </faultstring>
    <detail>
      <ns1:stackTrace xmlns:ns1="http://xml.apache.org/axis/">WSDLException:
faultCode=OTHER_ERROR: Can't find prefix for
'http://www.apple.com/webobjects/webservices/soap/'. Namespace
prefixes must be set on the Definition object using the addNamespace(...)
method.:
        at com.ibm.wsdl.util.xml.DOMUtils.getPrefix(Unknown Source)
        at com.ibm.wsdl.util.xml.DOMUtils.getQualifiedValue(Unknown Source)
        at com.ibm.wsdl.util.xml.DOMUtils.printQualifiedAttribute(Unknown
Source)
        at com.ibm.wsdl.xml.WSDLWriterImpl.printParts(Unknown Source)
        at com.ibm.wsdl.xml.WSDLWriterImpl.printMessages(Unknown Source)
        at com.ibm.wsdl.xml.WSDLWriterImpl.printDefinition(Unknown Source)
        at com.ibm.wsdl.xml.WSDLWriterImpl.writeWSDL(Unknown Source)
        at com.ibm.wsdl.xml.WSDLWriterImpl.getDocument(Unknown Source)
        at org.apache.axis.wsdl.fromJava.Emitter.emit(Emitter.java:267)
        at
org.apache.axis.providers.java.JavaProvider.generateWSDL(JavaProvider.java
:494)
        at
org.apache.axis.strategies.WSDLGenStrategy.visit(WSDLGenStrategy.java:72)
        at org.apache.axis.SimpleChain.doVisiting(SimpleChain.java:150)
        at org.apache.axis.SimpleChain.generateWSDL(SimpleChain.java:137)
        at
org.apache.axis.handlers.soap.SOAPService.generateWSDL(SOAPService.java:37
5)
        at

```

```
org.apache.axis.server.AxisServer.generateWSDL(AxisServer.java:499)
    at
com.webobjects.appserver._private.WOWebservice.performActionNamed(WOWebservice.java:352)
    at
com.webobjects.appserver._private.WOActionRequestHandler._handleRequest(WOActionRequestHandler.java:240)
    at
com.webobjects.appserver._private.WOActionRequestHandler.handleRequest(WOActionRequestHandler.java:142)
    at
com.webobjects.appserver._private.WOWebserviceRequestHandler.handleRequest(WOWebserviceRequestHandler.java:95)
    at
com.webobjects.appserver.WOApplication.dispatchRequest(WOApplication.java:1306)
    at
nz.co.lindesay.common.webobjects.LEWOApplication.dispatchRequest(LEWOApplication.java:537)
    at
nz.co.chong.cbw.webobjects.Application.dispatchRequest(Application.java:82)
    at
com.webobjects.appserver._private.WOWorkerThread.runOnce(WOWorkerThread.java:173)
    at
com.webobjects.appserver._private.WOWorkerThread.run(WOWorkerThread.java:254)
    at java.lang.Thread.run(Thread.java:552)
</ns1:stackTrace>
  </detail>
</soapenv:Fault>
```

```
</soapenv:Body>
</soapenv:Envelope>
```

Finally, if I look at the "com.webobjects.webservices.support.xml.WOGlobalIDDeserializer" class, it has a "byte[] data" instance variable so I can assume that at least some attempt has been made to deserialise temporary GID's.

Attempt at a Solution

The following is an attempt (it did not work) to write a custom serialiser/deserialiser for the AXIS environment in WebObjects such that these EOTemporaryGlobalID-s can be serialised. I think this was the point where I "dropped the SOAP" and moved to using JSON-RPC. However this will give anybody who is faced with this challenge a good starting point for further work in this area.

```
packagepackage nz.co.lindesay.common.webobjects;

// BSD LICENSE

import org.apache.axis.message.*;
import org.apache.axis.encoding.*;

import org.xml.sax.*;

import javax.xml.namespace.*;

import com.webobjects.foundation.*;
import com.webobjects.eocontrol.*;

/**
 * <P>WO's EOGlobalID deserializer is not able to handle temporary
 * global ID's and in fact throws some rather odd exceptions. For
 * this reason, I have developed my own deserialiser which will be
 * able to handle temporary global ID's.</P>
 *
 * <P><FONT color="#FF0000">This does not work.</FONT></P>
 */

public class LEWOWebservicesGlobalIDDeserializer extends
org.apache.axis.encoding.DeserializerImpl
{

    private static Integer DATA_HINT = new Integer(1);
    private static Integer PRIMARYKEY_HINT = new Integer(2);
    private static Integer ENTITYNAME_HINT = new Integer(3);

    protected NSData data = null;
    protected Object[] primaryKeys = null;
    protected String entityName = null;

    // -----

    public LEWOWebservicesGlobalIDDeserializer() { super(); }
```

```

// -----

    public void setChildValue(Object val, Object hint) throws
SAXException
    {
        if(hint.equals(DATA_HINT))
            data = (NSData) val;
        else
        {
            if(hint.equals(PRIMARYKEY_HINT))
                primaryKeys = (Object[]) val;
            else
            {
                if(hint.equals(ENTITYNAME_HINT))
                    entityName = (String) val;
            }
        }
    }

// -----

    public SOAPHandler onStartChild(
        String namespace,
        String localName,
        String prefix,
        Attributes attributes,
        DeserializationContext context) throws SAXException
    {
        DeserializerTarget dt = null;
        QName typeQName =
context.getTypeFromAttributes(namespace, localName, attributes);
        Deserializer dser =
context.getDeserializerForType(typeQName);

        // If no deserializer, use the base DeserializerImpl.
        if (null==dser)
            dser = new DeserializerImpl();

        if(localName.equals("data")) dt = new
DeserializerTarget(this, DATA_HINT);
        if(localName.equals("entityName")) dt = new
DeserializerTarget(this, ENTITYNAME_HINT);
        if(localName.equals("primaryKeys")) dt = new
DeserializerTarget(this, PRIMARYKEY_HINT);

        if(null==dt)
            throw new SAXException("the element '"+localName+"'
was encountered whilst deserialising an EOGlobalID and is not supported.");

        dser.registerValueTarget(dt);
        addChildDeserializer(dser);
    }

```

```

        return (SOAPHandler)dser;
    }

// -----

    public void onEndElement(
        String namespace,
        String localName,
        DeserializationContext context) throws SAXException
    {
        if(null!=data)
        {
            if((null!=entityName)|| (null!=primaryKeys))
                throw new SAXException("when deserialising
a temporary global id, the elements 'entityName' and 'primaryKey' should be
absent.");

                setValue(new
LEWOWebservicesTemporaryGlobalID(data.bytes()));
            }
            else
            {
                if(null!=entityName)
                {
                    if(null==primaryKeys)
                        throw new SAXException("when
deserialising a key global id, the element 'primarykey' should be
present.");

                    if(0==primaryKeys.length)
                        throw new SAXException("when
deserialising a key global id, the element 'primarykey' should contain one
or more key values.");

                setValue(EOKeyGlobalID.globalIDWithEntityName(entityName,primaryKeys));

                    }
                    else
                        throw new SAXException("when deserialising
a global id, either the data required for a key global id or a temporary
global id must be present.");
                }
            }
        }

// -----
// TEMPORARY GID SUBCLASS
// -----
// WO has the temporary GID exposed as protected
// which means that it is not possible to directly
// instantiate a temporary GID. For this reason,
// I have subclassed it in this inner class.
// -----

```

```
        public static class LEWOWebServicesTemporaryGlobalID extends
EOTemporaryGlobalID
    {

        public LEWOWebServicesTemporaryGlobalID(byte[]
globallyUniqueBytes) { super(globallyUniqueBytes); }

    }

// -----
```



```
}
```

```
package nz.co.lindesay.common.webobjects;

// BSD LICENSE

import com.webobjects.foundation.*;
import com.webobjects.appserver.*;
import com.webobjects.eocontrol.*;
import com.webobjects.webservices.support.xml.*;

import javax.xml.namespace.*;

import java.util.*;

public class LEWOWebServicesGlobalIDDeserializerFactory implements
org.apache.axis.encoding.DeserializerFactory
{

    private Set mechanisms = null;

// -----

    public static void registerWebServicesGlobalIDDeserialiserFactory()
    {
        LEWOWebServicesGlobalIDDeserializerFactory dserF = new
LEWOWebServicesGlobalIDDeserializerFactory();
        WOGlobalIDSerializerFactory serF = new
WOGlobalIDSerializerFactory();

        WOWebserviceRegistrar.registerFactoriesForClassWithQName(
            serF,
            dserF,
            EOGlobalID.class,
            WOSoapConstants.EOGLOBALID_QNAME);

        WOWebserviceRegistrar.registerFactoriesForClassWithQName(
            serF,
            dserF,
            EOGlobalID.class,
            WOSoapConstants.EOGLOBALID_QNAME_WEBSERVICESCORE_WORKAROUND);
    }

// -----

    public LEWOWebServicesGlobalIDDeserializerFactory() { super(); }

// -----
}
```

```
    public javax.xml.rpc.encoding.Deserializer getDeserializerAs(String
mechanismType)
        { return new LEWOWebServicesGlobalIDDSerializer(); }

// -----

    public Iterator getSupportedMechanismTypes()
    {
        if(null==mechanisms)
        {
            mechanisms = new HashSet();
            mechanisms.add(org.apache.axis.Constants.AXIS_SAX);
        }

        return mechanisms.iterator();
    }

// -----
```

}