

Articulation Transfer Clearinghouse **Implementation Guide**

for SendPescAcknowledgment

8/2/07

Table of Contents

INTRODUCTION.....	3
Project Identification	3
DOCUMENT CONTROL.....	4
Update History.....	4
ENVIRONMENTS	5
METHODS.....	6
JAVA IMPLEMENTATION DETAILS	7
.NET C# IMPLEMENTATION GUIDE.....	8
APPENDIX A: SENDPESCAKNOWLEDGMENT WSDL	9



Introduction

This implementation guide provides information needed to connect to and send a PESC Transcript Acknowledgment to the central ATC. It provides step-by-step insight into the creation of the payload at the client as well as the expected responses from the ATC.

PROJECT IDENTIFICATION		
Project Name	Project Number	Date Created
Articulation Transfer Clearinghouse	EN-009	05/22/2007
Sponsor	Program Manager	
Paula Compton, David Barber	Jim Ginzer	
Project Manager		
Jim Ginzer		

**Document Control**

This section contains document change control and distribution information.

UPDATE HISTORY			
Version	Date	Author	Description
0.1	05/22/07	Alex Leader – OBR	Authored
0.2	05/25/07	Alex Leader – OBR	Revisions following ATC pilot discussion (CINC, CNST, NEOUCOM, SNCL, WSUN)
0.3	06/28/07	Alex Leader – OBR	Update after development completed.



Environments

OBR provides both a testing and production implementation of SendPescAcknowledgment for your convenience. Clients should develop against the test version of this service, then migrate their code accordingly to use the production version of the service. The following URLs correspond to these instances:

Environment	WSDL URL
User Acceptance Testing (UAT)	https://uat-atc.regents.ohio.gov/ccx/SendPescAcknowledgment?wsdl
Production (PROD)	https://atc.regents.ohio.gov/ccx/SendPescAcknowledgment?wsdl



Methods

Method Name: SendPescAcknowledgment

Description

Sends a PESC acknowledgment to OBR for delivery to the source institution.

Request

Username – OBR supplied institutional username

Password – OBR supplied institutional password

Data – compressed and base64 encoded data packet (xml acknowledgment)

Response

DocumentId – Original DocumentID specified in acknowledgment.

TrackingId – OBR assigned tracking ID for reference to the ATC.

Fault

DetailMessage – Detailed description of thrown error message.



Java Implementation Details

Using the WSDL found in Appendix A, a top-down development approach can be implemented. Within your IDE of choice, create a project from the WSDL or set a reference to the WSDL. Your IDE should assist you in creating the required code to call the web service. The targetNamespace and soap:address location values will require modification dependent on the environment being connected to at OBR.

Assumptions:

1. IDE of your choice
2. Framework for Web Service development/deployment

Prerequisites:

1. Point your browser to <https://uat-atc.regents.ohio.gov/ccx/SendPescAcknowledgment?wsdl>.
2. The acknowledgment will need to be Base64 encoded. One way to accomplish this is by using Commons Codec from <http://jakarta.apache.org/commons/codec/>. This library provides Base64 encoding/decoding functionality

Implementation Steps:

1. The XML acknowledgment for the SendPescAcknowledgment service must be compressed utilizing zlib. This can be accomplished using `java.util.zip.Deflater`
2. Base64 encode the acknowledgment using `org.apache.commons.codec.binary.Base64` from the Commons Codec library.
3. Call the `SendPescAcknowledgment` method passing in the compressed and encoded acknowledgment as the 'Data' element in the `SendPescAcknowledgmentRequest` message. Use the OBR supplied username and password for the 'Username' and 'Password' elements of the `SendPescAcknowledgmentRequest` message.
4. After calling the `SendPescAcknowledgment` service, the web service client will receive a SOAP response or a SOAP fault. The response is of type `SendPescAcknowledgmentResponse` which contains both a `TrackingID` and a `DocumentID`. These values may be used by the institution to store an ATC reference to the acknowledgment within their SIS system. Additionally, if a SOAP fault or other exception is returned, the web service client should catch that exception and take the appropriate action.



.Net C# Implementation Guide

Using the WSDL found in Appendix A, a top-down development approach can be implemented. Within your IDE of choice, create a project from the WSDL or set a reference to the WSDL. Your IDE should assist you in creating the required code to call the web service. The targetNamespace and soap:address location values will require modification dependent on the environment being connected to at OBR.

Assumptions:

1. Visual Studio 2005
2. .Net 2.0
3. C#

Prerequisites:

1. Obtain zlib.net.dll from <http://www.componentace.com/ZLIB.NET>.
2. Point your browser to <https://uat-atc.regents.ohio.gov/ccx/SendPescAcknowledgment?wsdl>.

Implementation Steps:

1. The acknowledgment for the SendPescAcknowledgment service must be compressed utilizing zlib. Create a reference to the DLL in your project. Compress the acknowledgment.
2. Base64 encode the acknowledgment (`System.Convert.ToBase64String`).
3. Call the SendPescAcknowledgment method passing in the compressed and encoded acknowledgment as the 'Data' element in the SendPescAcknowledgmentRequest message. Use the OBR supplied username and password for the 'Username' and 'Password' elements of the SendPescAcknowledgmentRequest message.
4. After calling the SendPescAcknowledgment service, the web service client will receive a SOAP response or a SOAP fault. The response is of type SendPescAcknowledgmentResponse which contains both a TrackingID and a DocumentID. These values may be used by the institution to store an ATC reference to the acknowledgment within their SIS system. Additionally, if a SOAP fault or other exception is returned, the web service client should catch that exception and take the appropriate action.



Appendix A: SendPescAcknowledgment WSDL

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:tns="http://www.regents.ohio.gov/SendPescAcknowledgment/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:atc="http://www.regents.ohio.gov/SendPescAcknowledgment/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" name="SendPescAcknowledgment"
targetNamespace="http://www.regents.ohio.gov/SendPescAcknowledgment/">
  <wsdl:types>
    <xsd:schema xmlns:tns="http://www.regents.ohio.gov/SendPescAcknowledgment/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.regents.ohio.gov/SendPescAcknowledgment/">
      <xsd:element name="SendPescAcknowledgmentResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="DocumentId" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="TrackingId" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="SendPescAcknowledgmentRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="0" name="Username" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="Password" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="Data" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="SendPescAcknowledgmentFault">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="DetailMessage" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="SendPescAcknowledgmentRequest">
    <wsdl:part name="SendPescAcknowledgmentRequest" element="tns:SendPescAcknowledgmentRequest"/>
  </wsdl:message>
  <wsdl:message name="SendPescAcknowledgmentResponse">
    <wsdl:part name="SendPescAcknowledgmentResponse" element="tns:SendPescAcknowledgmentResponse"/>
  </wsdl:message>
  <wsdl:message name="SendPescAcknowledgmentFault">
    <wsdl:part name="SendPescAcknowledgmentFault" element="tns:SendPescAcknowledgmentFault"/>
  </wsdl:message>
  <wsdl:portType name="SendPescAcknowledgment">
    <wsdl:operation name="SendPescAcknowledgment">
      <wsdl:input message="tns:SendPescAcknowledgmentRequest"/>
      <wsdl:output message="tns:SendPescAcknowledgmentResponse"/>
      <wsdl:fault name="SendPescAcknowledgmentFault" message="tns:SendPescAcknowledgmentFault"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="SendPescAcknowledgmentSOAP" type="tns:SendPescAcknowledgment">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="SendPescAcknowledgment">
      <soap:operation soapAction="http://www.regents.ohio.gov/SendPescAcknowledgment/SendPescAcknowledgment"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="SendPescAcknowledgmentFault">
        <soap:fault name="SendPescAcknowledgmentFault"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="SendPescAcknowledgment">
    <wsdl:port name="SendPescAcknowledgmentSOAP" binding="tns:SendPescAcknowledgmentSOAP">
      <soap:address location="https://uat-atc.regents.ohio.gov/ccx/SendPescAcknowledgment"/>
    </wsdl:port>
  </wsdl:service>

```



`</wsdl:service>`
`</wsdl:definitions>`