# Electronic Records Express Web Services (EREWS)

# Client Development Guide

JULY 2, 2007

## REVISION HISTORY

| Date | Version | Description | Author(s) |
|---|---|---|---|
| 07/02/2007 | 0.1 | Preliminary Draft | EREWS Team |
| | | | |
| | | | |

# Table of Contents

## Version 0.1

1.0 INTRODUCTION ........ 4
2.0 ERE WEB SERVICES & WSDL........ 4
3.0 Details of WSDL file:........ 5
4.0 Security........ 7
5.0 Sample Implementation of EREWS Client Application (JAVA)........ 8
6.0 Sample Implementation of EREWS Client Application (.NET)........ 10
7.0 References........ 12

Appendix A
Appendix B

# 1.0 INTRODUCTION

Electronic Records Express Web Services (EREWS) as a solution is to provide SSA with a high volume Internet transport to receive electronic evidence from SSA's business partners, leveraging the Internet and associated industry standard protocols. EREWS provides two services (Ping and file upload) and does not have a user interface. Users have to develop their own client software to integrate with the EREWS SSA systems. This documents helps you to provide a description of the services provided and the connection procedures for connecting to the ERE Web Services.

# 2.0 ERE WEB SERVICES & WSDL

ERE Web Services is described in a Web Services Description Language (WSDL). The WSDL file is available for the Web Services client users/developers to implement their own client applications. Note that some information, such as the http header information, security token, and attachments cannot be described in WSDL.

ERE Web Services Client application can be developed as a standalone Java application, Java EE Web Application or a .NET application developed in C# or any other platform. In the following discussion, we will focus on the standalone Java and .NET clients only.

Let's start with the URL of WSDL – an XML file.

The URL's:

Validation: https://valws.ssa.gov/EREWS/services/EREWSSOAP?wsdl

Integration: https://intws.ssa.gov/EREWS/services/EREWSSOAP?wsdl

Production: https://ws.ssa.gov/EREWS/services/EREWSSOAP?wsdl

Depending on the management decisions and time frames you need to proceed and connect to the appropriate URL's.

***Required items to while connecting to EREWS:***

1. URL
2. PIN/Password (provided by SSA)
3. Develop your own Client Software (not supported by SSA)

If you have any problems with the WSDL file please contact the EREWS team they can help you with the WSDL file. A sample WSDL file is appended to the end of this document.

## 3.0 Details of WSDL file:

**Service Endpoint Interface (SEI)**

Namespace: **http://ssa.gov/ere**
Interface Name: **EREWS_PortType**

Currently EREWS provides two types of services:

### 1. Ping:

To check whether the server is up and running. The return parameter is a "date + timestamp" in a string format.

***Method Signature PRequest for ping function:***

**PRequest(PingRequest) : PingResponse**

The following are the definitions for PingRequest and PingResponse

PingRequest:

- requestparam : String

PingResponse:

- responseparam : String

The *requestparam* is a string to determine whether the web service is up and running. A string value is sent to the web service.

The *responseparam* returns the date-time stamp in a string format, notifying the client the server is up and running successfully.

### 2. SubmitBulk:

Files are uploaded using the submitBulk method. A response is expected depending on the parameters passed to the Web Service.

***Method Signature submitBulk for uploading files:***

**submitBulk (AttachmentInfo attachment, byte[] attachmentFile) : ReturnInfo**

The following are the definitions for AttachmentInfo and ReturnInfo.

AttachmentInfo:

- filename : string
- fileSize : long
- checksum : string

attachmentFile : Attachment as byte array

The *filename* should be 32 characters in length. These files are stored on z/OS hierarchical file system (HFS). The *filename* should be in a DMA specified format otherwise it will not be processed to the DMA.

The *fileSize* is in bytes and is to verify whether the file uploaded is same as received on the server side. It should not be a zero byte file.

The *checksum* is the MD5 (Message-Digest algorithm 5) checksum string of 32 lowercase hexadecimal digits for the uploaded ZIP file. Note that neither "*0x*" prefix nor "*h*" suffix is included in *checksum* string. More information on MD5 can be accessed at http://en.wikipedia.org/wiki/MD5.

MD5 digests have been widely used in the software world to provide some assurance that a downloaded file has not been altered. We use it for the upload file integrity verification.

ERE Web Services will calculate the byte count and the MD5 checksum of the file it receives and compare those values to that which are sent in the SOAP message. An error code will be returned if either of these two values does not agree. Should this occur, we know the uploaded file is corrupted, incomplete, or otherwise altered during the transmission.

*attachmentFile*: The *attachmentFile* is a byte array of the attachment.

ReturnInfo:

- confirmationCode : string
- status : string
- receiptDate : string
- assignedFileName : string
- receiptFileSize : long
- receiptFileChecksum : string

The *confirmationCode* is sent by the server and contains the confirmation number for the uploaded file. This is used to keep track of the files.

The **status** can be any of the following:

SUCCESS – successful upload

CORRUPT_FILE – Failed in MD5 checksum or File size or Filename

SERVER_ERROR – File I/O Error or Database Connection Error

TRANSPORT_ERROR – Message Context not available

The *receiptDate* is a date-time stamp calculated by the server and sent in string format.

The *assignedFileName* is the filename assigned to the file when received by the server.

The *receiptFileSize* is the file size in bytes that is received by the server. This is to verify whether the actual file sent is the one that is received on the server.

The *receiptFileChecksum* is the MD5 checksum calculated on the received file by the server.

**Note:**

1. If you haven't received a *confirmationCode* you need to upload the file again.
2. If you haven't received the *status* as SUCCESS you need to upload the file again
3. The files having the *confirmationCode* and *status* as SUCCESS are the only files that are processed.

The Web Services client applications will attach the ZIP file in the SOAP Message by using inline binary attachment fromat.

## 4.0 Security:

Medical information transmitted over the internet is required to be secure. To guarantee the confidentiality of the medical information, HTTPS (HTTP over SSL) is used. The service endpoint (URL address of EREWS) to which the client application will connect to a secured server. The use of SSL will ensure the confidentiality of the data being transmitted.

Also we use the WS-Security standard and to authenticate the user. We follow the OASIS security standards and more information can be available at http://www.oasis-open.org/specs/index.php#wssv1.0
When implementing the client for Web Services, the users should use the security with SSL and basic authentication. PIN & Password must be set in the "UsernameToken" in the SOAP Header and is used to authenticate the user. It is provided to you once SSA approves your request.

A sample request with the security header and Username token is provided in the Appendix at the end of this document

# 5.0 Sample Implementation of EREWS Client Application (JAVA)

We used Apache Axis1.4 and Java1.4 for generating the client.

**Instructions:**

1. First step is to obtain the WSDL file from SSA. WSDL – Web Services Definition Language. This XML file describes the Service Endpoint Interface.

2. Generate SOAP proxy classes using WSDL2Java code generator (provided with Apache Axis). The generated classes will be in the package gov.ssa.ere.client.

3. Develop StubClient.java which is the Proxy class.

4. Make sure the following files are in your classpath. They are developed by SSA as a sample to be used as a reference. You may implement additional requirements for monitoring/tracking on the client side as needed.

PWCallback.java – used to implement WS-Security
CheckSumUtil.java – generate MD5 (or SHA) hash string. For the current implementation, we will use MD5.
StubClient.java – invokes ERE Web Service

**Attachment Support**

Attachment is included in the SOAP request using inline binary format. To include the attachment in the SOAP request first convert the attachment into a byte array and then include it in the *attachmentFile* element.

**Properties Dependencies:**

These properties files are required to be in the classpath, in expanded format (not in a zip or jar).

client-config.wsdd – configuration properties for SOAP client.
For this release make sure the ("mustUnderstand" attribute is set to false)

**Third Party Libraries**

These libraries are used for creating the SOAP message, packaging the attachment, and establishing an HTTP connection.

**Apache Axis 1.4**:
axis-ant.jar
axis-schema.jar
axis.jar
commons-discovery-0.2.jar
commons-logging-1.0.4.jar
jaxrpc.jar
log4j-1.2.8.jar
saaj.jar
wsdl4j-1.5.1.jar

**Javabeans Activation Framework 1.0.2**:
activation.jar

**JavaMail API 1.3.3**:
mail.jar

**WSS4J 1.5**:
wss4j-1.5.0.jar

**Apache XML Security 1.2.1**:
xmlsec-1.2.1.jar
xalan.jar

Compile StubClient.java (or similar), utility classes and the Axis generated classes.

**Application Deployment/Implementation**

Create a zip file for attachment to the SOAP message. The ZIP file should be in the DMA-accepted format.

run the client class (with the appropriate jar's in the classpath)

```
C:\java gov.ssa.ere.client.StubClient
```

***A sample JAVA toolkit can be provided to you upon your request.***

# 6.0 Sample Implementation of EREWS Client Application (.NET)

**Instructions:**

Our environment: Microsoft .NET 2.0, Visual C# 2005 Express Edition, WSE 2.0 but the source code is easily translated to Visual Basic.

1. Obtain copy of WSDL and save it locally. (rename EREWS.txt to EREWS.wsdl)
2. In that directory, run wsdl.exe to generate the proxy class: "C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\Bin\wsdl" /l:CS /protocol:SOAP .\EREWS.wsdl
3. After generating the proxy, replace the Unqualified attributes as Qualified attribute. (For getting the Qualified attributes to serialize properly in the .Net here is the fix http://support.microsoft.com/kb/327071 )
4. Make sure that your proxy class inherits from Microsoft.Web.Services2.WebServicesClientProtocol. It will use the default System.Web.Services.Protocols.SoapHttpClientProtocol (which does not contain a definition for RequestSoapContext).
5. Edit the proxy for proper namespace attributes
    a. Add the attributes Namespace="" and IsNullable=false to the parameters for submitBulk and PRequest methods. Example: [System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSchemaForm.Qualified, **Namespace="", IsNullable=false**)] AttachmentInfo attachmentInfo,
    b. Remove the [System.Xml.Serialization.XmlTypeAttribute(Namespace="http://ssa.gov/ere")] for the methods AttachmentInfo ,ReturnInfo, PingRequest and PingResponse
6. Compile the proxy class with csc.exe: "C:\WINNT\Microsoft.NET\Framework\v2.0.50727\csc" /t:library /r:System.Web.Services.dll /r:Microsoft.Web.Services2.dll /r:System.Xml.dll EREWS.cs
7. Create a new console application in Visual C#.
8. In solution explorer, add to the project a reference, and navigate to the EREWS.dll created in step 4.
9. In solution explorer, add a web reference – https://valws.ssa.gov/EREWS/services/EREWSSOAP?wsdl and click “Go.” Click OK for both pop-up boxes concerning the certificates.
10. Create the client application source file similar to below. Our example is in Visual C# 2005. The executable has one command line argument, ping or submitBulk. If running the submitBulk function, be sure the FileStream object points to a valid and readable file.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using Microsoft.Web.Services2;
using Microsoft.Web.Services2.Security;

class EREWSClient
{
   static void Main(string[] args)
   {
      Microsoft.Web.Services2.Security.Tokens.UsernameToken userTok;
      string agentId = "USERNAME";
      string authCode = "PASSWORD";
      EREWS eREWS = new EREWS();
```

```
        userTok = new Microsoft.Web.Services2.Security.Tokens.UsernameToken(agentId,
authCode, Microsoft.Web.Services2.Security.Tokens.PasswordOption.SendPlainText);
        eREWS.RequestSoapContext.Security.Tokens.Add(userTok);



eREWS.RequestSoapContext.Security.MustUnderstand = false;
eREWS.Timeout = -1;
AttachmentInfo attachmentInfo = new AttachmentInfo();
attachmentInfo.fileName = "text.txt";
attachmentInfo.fileSize = 15;
attachmentInfo.checksum = "e0299c93eb237e28fdfffd85943ece96";
FileStream fs = new FileStream("D:\\files\\text.txt", FileMode.Open);
AttachmentInfo info = new AttachmentInfo();
byte[] bytes = new byte[fs.Length];
fs.Read(bytes, 0, (int)fs.Length);
fs.Close();
ReturnInfo submitBulkResult = eREWS.submitBulk(attachmentInfo, bytes);
Console.WriteLine("receiptFileChecksum=" + submitBulkResult.receiptFileChecksum);
Console.WriteLine("assignedFileName=" + submitBulkResult.assignedFileName);
Console.WriteLine("confirmationCode=" + submitBulkResult.confirmationCode);
Console.WriteLine("receiptDate=" + submitBulkResult.receiptDate);
Console.WriteLine("receiptFileSize=" + submitBulkResult.receiptFileSize);
Console.WriteLine("status=" + submitBulkResult.status);
Console.ReadLine();
/*
      PingRequest pingReq = new PingRequest();
      pingReq.requestParam = "rdgf";
      PingResponse pRequestResult = eREWS.PRequest(pingReq);
      Console.WriteLine("status" + pRequestResult.responseParam);
      Console.ReadLine();
      */
      }
}
```

***A sample .NET tool and a proxy file can be provided to you upon your request.***

## 7.0 References:

**MD5 checksum:**
http://en.wikipedia.org/wiki/MD5

**SOAP Implementation:**
http://ws.apache.org/axis/

**Web Services Security:**

Johan Danforth's tutorial for using WSS4J with Axis:
http://weblogs.asp.net/jdanforth/archive/2005/01/16/354060.aspx

Apache's WSS4J/Axis page:
http://ws.apache.org/wss4j/axis.html

OASIS WS-Security
http://www.oasis-open.org/specs/index.php#wssv1.0

## Appendix A:

**WSDL:**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<wsdl:definitions targetNamespace="http://ssa.gov/ere"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://ssa.gov/ere" xmlns:intf="http://ssa.gov/ere"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
    <wsdl:types>
        <schema targetNamespace="http://ssa.gov/ere"
xmlns="http://www.w3.org/2001/XMLSchema">
            <element name="submitBulk">
                <complexType>
                    <sequence>
                        <element name="attachmentInfo"
type="impl:AttachmentInfo"/>
                        <element name="attachmentFile"
type="xsd:base64Binary"/>
                    </sequence>
                </complexType>
            </element>
            <complexType name="AttachmentInfo">
                <sequence>
                    <element name="fileName" type="xsd:string"/>
                    <element name="fileSize" type="xsd:long"/>
                    <element name="checksum" type="xsd:string"/>
                </sequence>
            </complexType>
            <element name="submitBulkResponse">
                <complexType>
                    <sequence>
                        <element name="submitBulkReturn"
type="impl:ReturnInfo"/>
                    </sequence>
                </complexType>
            </element>
            <complexType name="ReturnInfo">
                <sequence>
                    <element name="confirmationCode" nillable="true"
type="xsd:string"/>
                    <element name="status" nillable="true"
type="xsd:string"/>
                    <element name="receiptDate" nillable="true"
type="xsd:string"/>
                    <element name="assignedFileName" nillable="true"
type="xsd:string"/>
                    <element name="receiptFileSize" nillable="true"
type="xsd:long"/>
                    <element name="receiptFileChecksum"
nillable="true" type="xsd:string"/>
                </sequence>
            </complexType>
            <element name="PRequest">
                <complexType>
                    <sequence>
                        <element name="pingReq"
type="impl:PingRequest"/>
                    </sequence>
                </complexType>
            </element>
            <complexType name="PingRequest">
                <sequence>
                    <element name="requestParam" type="xsd:string"/>
                </sequence>
            </complexType>
            <element name="PRequestResponse">
                <complexType>
                    <sequence>
                        <element name="PRequestReturn"
type="impl:PingResponse"/>
                    </sequence>
                </complexType>
            </element>
            <complexType name="PingResponse">
                <sequence>
                    <element name="responseParam"
type="xsd:string"/>
                </sequence>
            </complexType>
        </schema>
    </wsdl:types>
    <wsdl:message name="submitBulkResponse">
        <wsdl:part element="impl:submitBulkResponse"
name="parameters"/>
    </wsdl:message>
    <wsdl:message name="PRequestResponse">
        <wsdl:part element="impl:PRequestResponse"
name="parameters"/>
    </wsdl:message>
    <wsdl:message name="submitBulkRequest">
        <wsdl:part element="impl:submitBulk" name="parameters"/>
    </wsdl:message>
    <wsdl:message name="PRequestRequest">
        <wsdl:part element="impl:PRequest" name="parameters"/>
    </wsdl:message>
    <wsdl:portType name="EREWS">
        <wsdl:operation name="submitBulk">
            <wsdl:input message="impl:submitBulkRequest"
name="submitBulkRequest"/>
            <wsdl:output message="impl:submitBulkResponse"
name="submitBulkResponse"/>
        </wsdl:operation>
        <wsdl:operation name="PRequest">
            <wsdl:input message="impl:PRequestRequest"
name="PRequestRequest"/>
            <wsdl:output message="impl:PRequestResponse"
name="PRequestResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="EREWSSOAPSoapBinding" type="impl:EREWS">
        <wsdlsoap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="submitBulk">
            <wsdlsoap:operation soapAction="http://ssa.gov/ere"/>
            <wsdl:input name="submitBulkRequest">
                <wsdlsoap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="submitBulkResponse">
                <wsdlsoap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="PRequest">
            <wsdlsoap:operation soapAction="http://ssa.gov/ere"/>
            <wsdl:input name="PRequestRequest">
                <wsdlsoap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="PRequestResponse">
                <wsdlsoap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="EREWS">
        <wsdl:port binding="impl:EREWSSOAPSoapBinding"
name="EREWSSOAP">
            <wsdlsoap:address
location="https://ws.ssa.gov/EREWS/services/EREWSSOAP"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

## Appendix B:

### HTTP header with basic Authentication / Password

```
POST /EREWS/services/EREWSSOAP HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.4
Host: sy51.sspf.ssa.gov:8148
```

### Sample SOAP request sent by ERE Web Services Java Client:

```
<?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:Header><wsse:Security xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd"><wsse:UsernameToken wsu:Id="UsernameToken-24962279"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"><wsse:Username>USERNAME</wsse:Username><wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-
1.0#PasswordText">PASSWORD</wsse:Password></wsse:UsernameToken></wsse:Se
curity></soapenv:Header><soapenv:Body><submitBulk
xmlns="http://ssa.gov/ere"><attachmentInfo xmlns=""><fileName>
SDS001.R070411.AAAA05.DS270M.ZIP
</fileName><fileSize>15</fileSize><checksum>e0299c93eb237e28fdfffd85943ece96</c
hecksum></attachmentInfo><attachmentFile
xmlns="">c21hbGwgdGV4dCBmaWxl</attachmentFile></submitBulk></soapenv:Body><
/soapenv:Envelope>
```

### Sample SOAP request sent by ERE Web Services .Net Client:

```
<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"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-
1.0.xsd"><soap:Header><wsa:Action>http://ssa.gov/ere</wsa:Action><wsa:MessageID>
uuid:1580ab1a-d5e2-47c0-a55f-
4642c8b3d553</wsa:MessageID><wsa:ReplyTo><wsa:Address>http://schemas.xmlsoap
.org/ws/2004/03/addressing/role/anonymous</wsa:Address></wsa:ReplyTo><wsa:To>htt
p://10.32.131.15:9080/EREWS/services/EREWSSOAP</wsa:To><wsse:Security><wsu:
Timestamp wsu:Id="Timestamp-ba17974d-3c09-4b44-b880-
d68a9fbd6dc3"><wsu:Created>2007-04-
05T19:13:24Z</wsu:Created><wsu:Expires>2007-04-
05T19:18:24Z</wsu:Expires></wsu:Timestamp><wsse:UsernameToken
wsu:Id="SecurityToken-73ff1dab-7e8e-49cb-876f-
ef852e132bd6"><wsse:Username>USERNAME</wsse:Username><wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-
1.0#PasswordText">PASSWORD</wsse:Password><wsse:Nonce>BGkNDQui0EAyQj2
```

Xd38x2g==</wsse:Nonce><wsu:Created>2007-04-
05T19:13:24Z</wsu:Created></wsse:UsernameToken></wsse:Security></soap:Header>
<soap:Body><submitBulk xmlns="http://ssa.gov/ere"><attachmentInfo
xmlns=""><fileName> SDS001.R070411.AAAA05.DS270M.ZIP
</fileName><fileSize>15</fileSize><checksum>
e0299c93eb237e28fdfffd85943ece96</checksum></attachmentInfo><attachmentFile
xmlns="">c21hbGwgdGV4dCBmaWxl</attachmentFile></submitBulk></soap:Body>
</soap:Envelope>

**Sample SOAP response sent by ERE Web Services.**

<?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><submitBulkResponse
xmlns="http://ssa.gov/ere"><submitBulkReturn><confirmationCode>111EC84C8F87BC8
4</confirmationCode><status>SUCCESS</status><receiptDate>04/13/2007
15:56:36.940 PM</receiptDate><assignedFileName>111EC84C8F87BC84.
SDS001.R070411.AAAA05.DS270M.ZIP
</assignedFileName><receiptFileSize>15</receiptFileSize><receiptFileChecksum>e029
9c93eb237e28fdfffd85943ece96</receiptFileChecksum></submitBulkReturn></submitBu
lkResponse></soapenv:Body></soapenv:Envelope>