



Sample Codes | 1.2

ACHWorks-SOAP Ver 4.0 Sample Codes

Table of Contents

SAMPLE CLIENT CODES	3
A2.1 PHP	3
A2.2 VISUAL BASIC (VB.NET)	12
A2.3 C#	20
A2.4 JAVA	28

Sample Client Codes

The sample codes below are written for the most popular and recent development platforms namely PHP, C#, Visual Basic (VB.NET), ASP.NET and JAVA.

Brief descriptions are provided to provide ease in understanding the required input and output and their formats.

Please contact us if you want a digital copy of the sample codes. See Section 4 for contact information.

A2.1 PHP (Version 5)

connectioncheck.php

```
<?php
//ACHWORKS-SOAP Ver4.0 ConnectionCheck (Connection Check)
//company info
class CompanyInfo {
    public $SSS;
    public $LocID;
    public $Company;
    public $CompanyKey;
}

$mycompanyinfo = new CompanyInfo;
$mycompanyinfo -> SSS = "TST";
$mycompanyinfo -> LocID = "9505";
$mycompanyinfo -> Company = "THAT COMPANY";
$mycompanyinfo -> CompanyKey = "RICO";

$myclient = new SoapClient("http://tstsvr.achworks.com/dnet/achws.asmx?WSDL");

//Important: use InpCompanyInfo in the soap stru not CompanyInfo
$myresult = $myclient->ConnectionCheck(array("InpCompanyInfo"=>$mycompanyinfo));

print($myresult->ConnectionCheckResult);
?>
```

sendachtrans.php

```
<?php
//ACHWORKS-SOAP Ver3.0 SendACHTrans (Sends a single record)
//3.25.2010 - rico pamplona, rpamplonaATachworksDOTcom
//company info
class CompanyInfo {
    public $SSS;
    public $LocID;
    public $Company;
    public $CompanyKey;
}

class ACHTransRecord {
    public $SSS;
    public $LocID;
    public $FrontEndTrace;
    public $OriginatorName;
    public $TransactionCode;
    public $CustTransType;
    public $CustomerID;
    public $CustomerName;
    public $CustomerRoutingNo;
    public $CustomerAcctNo;
    public $CustomerAcctType;
    public $TransAmount;
    public $CheckOrCustID;
    public $CheckOrTransDate;
    public $EffectiveDate;
    public $Memo;
    public $OpCode;
    public $AccountSet;
}

//CompanyInfo
$mycompanyinfo = new CompanyInfo;
$mycompanyinfo -> SSS = "TST";
$mycompanyinfo -> LocID = "9505";
$mycompanyinfo -> Company = "THAT COMPANY";
$mycompanyinfo -> CompanyKey = "RICO";

//ACHTransRecord
$myachtransrecord = new ACHTransRecord;
$myachtransrecord -> SSS = "TST"; //T$$ assigned
$myachtransrecord -> LocID = "9505"; //T$$ assigned
$myachtransrecord -> FrontEndTrace = "W9000123310454"; //impt!!!Needs a unique value each record per LocID
$myachtransrecord -> OriginatorName = "MYCOMPANY";
$myachtransrecord -> TransactionCode = "PPD"; //PPD, CCD, WEB, TEL, RCK, etc.
$myachtransrecord -> CustTransType = "D"; //D for debit, C for credit
$myachtransrecord -> CustomerID = "CustID12235";
$myachtransrecord -> CustomerName = "DOE, JOHNNY";
$myachtransrecord -> CustomerRoutingNo = "987654320";
$myachtransrecord -> CustomerAcctNo = "0000388836291";
```

```

$myachtransrecord -> CustomerAcctType = "C"; //C for checking, S for Savings
$myachtransrecord -> TransAmount = 100.75;
$myachtransrecord -> CheckOrCustID = "9166388811";
$myachtransrecord -> CheckOrTransDate = '2010-03-24'; //include leading zero for mm and dd e.g. 01 for Jan
$myachtransrecord -> EffectiveDate = '2010-03-24'; //include leading zero for mm and dd e.g. 01 for Jan
$myachtransrecord -> Memo = "Payment";
$myachtransrecord -> OpCode = "S"; //S for Single, R for recurring
$myachtransrecord -> AccountSet = "1"; //T$$ assigned

```

```

//SOAP call - test server
$myclient = new SoapClient("http://tstsvr.achworks.com/dnet/achws.asmx?WSDL");
$myresult = $myclient->SendACHTrans(array("InpCompanyInfo"=>$mycompanyinfo,"InpACHTransRecord"=>$myachtransrecord))-
    >SendACHTransResult;

```

```

//print status and details
print($myresult->Status . " , " . $myresult->Details . "<br>");

```

```

//print errors if there is any
foreach ($myresult->Errors->string as $myerror) {
print($myerror . "<br>");
}
?>

```

sendachtransbatch.php

```

<?php
//ACHWORKS-SOAP Ver3.0 SendACHTransBatch (Sends multiple records)
//3.25.2010 - rico pamplona, rpamplonaATachworksDOTcom
//company info
class CompanyInfo {
    public $SSS;
    public $LocID;
    public $Company;
    public $CompanyKey;
}

class ACHTransRecord {
    public $SSS;
    public $LocID;
    public $FrontEndTrace;
    public $OriginatorName;
    public $TransactionCode;
    public $CustTransType;
    public $CustomerID;
    public $CustomerName;
    public $CustomerRoutingNo;
    public $CustomerAcctNo;
    public $CustomerAcctType;
    public $TransAmount;
    public $CheckOrCustID;
    public $CheckOrTransDate;
    public $EffectiveDate;
    public $Memo;
}

```

```
    public $OpCode;
    public $AccountSet;
}
```

```
class ACHFile {
    public $SSS;
    public $LocID;
    public $ACHFileName;
    public $TotalNumRecords;
    public $TotalDebitRecords;
    public $TotalDebitAmount;
    public $TotalCreditRecords;
    public $TotalCreditAmount;
    public $ACHRecords;
}
```

```
//CompanyInfo
$mycompanyinfo = new CompanyInfo;
$mycompanyinfo -> SSS = "TST";
$mycompanyinfo -> LocID = "9505";
$mycompanyinfo -> Company = "THAT COMPANY";
$mycompanyinfo -> CompanyKey = "RICO";
```

```
//sample ACHTransRecord 1,2 & 3 (2 debits and 1 credit)
//1
$myachtransrecord1 = new ACHTransRecord;
$myachtransrecord1 -> SSS = "TST"; //T$$ assigned
$myachtransrecord1 -> LocID = "9505"; //T$$ assigned
$myachtransrecord1 -> FrontEndTrace = "W3000123456"; //impt!!!Needs a unique value each record per LocID
$myachtransrecord1 -> OriginatorName = "MYCOMPANY";
$myachtransrecord1 -> TransactionCode = "PPD"; //PPD, CCD, WEB, TEL, RCK, etc.
$myachtransrecord1 -> CustTransType = "D"; //D for debit, C for credit
$myachtransrecord1 -> CustomerID = "CustID12235";
$myachtransrecord1 -> CustomerName = "DOE, JOHNNY";
$myachtransrecord1 -> CustomerRoutingNo = "987654320";
$myachtransrecord1 -> CustomerAcctNo = "04038883674";
$myachtransrecord1 -> CustomerAcctType = "C"; //C forchecking, S for Savings
$myachtransrecord1 -> TransAmount = 100.25;
$myachtransrecord1 -> CheckOrCustID = "9166388811";
$myachtransrecord1 -> CheckOrTransDate = '2010-03-16'; //include leading zero for mm and dd e.g. 01 for Jan
$myachtransrecord1 -> EffectiveDate = '2010-03-16'; //include leading zero for mm and dd e.g. 01 for Jan
$myachtransrecord1 -> Memo = "Payment";
$myachtransrecord1 -> OpCode = "S"; //S for Single, R for Recurring
$myachtransrecord1 -> AccountSet = "1"; //T$$ assigned
```

```
//2
$myachtransrecord2 = new ACHTransRecord;
$myachtransrecord2 -> SSS = "TST"; //T$$ assigned
$myachtransrecord2 -> LocID = "9505"; //T$$ assigned
$myachtransrecord2 -> FrontEndTrace = "W3000123457"; //impt!!!Needs a unique value each record per LocID
$myachtransrecord2 -> OriginatorName = "MYCOMPANY";
$myachtransrecord2 -> TransactionCode = "PPD"; //PPD, CCD, WEB, TEL, RCK, etc.
$myachtransrecord2 -> CustTransType = "D"; //D for debit, C for credit
$myachtransrecord2 -> CustomerID = "CustID12235";
```

```

$myachtransrecord2 -> CustomerName = "DOE, JANET";
$myachtransrecord2 -> CustomerRoutingNo = "987654320";
$myachtransrecord2 -> CustomerAcctNo = "0000388836291";
$myachtransrecord2 -> CustomerAcctType = "C"; //C forchecking, S for Savings
$myachtransrecord2 -> TransAmount = 500.25;
$myachtransrecord2 -> CheckOrCustID = "9166388811";
$myachtransrecord2 -> CheckOrTransDate = '2010-03-16'; //include leading zero for mm and dd e.g. 01 for Jan
$myachtransrecord2 -> EffectiveDate = '2010-03-16'; //include leading zero for mm and dd e.g. 01 for Jan
$myachtransrecord2 -> Memo = "Payment";
$myachtransrecord2 -> OpCode = "S"; //S for Single, R for Recurring
$myachtransrecord2 -> AccountSet = "1"; //T$$ assigned

```

```

//3
$myachtransrecord3 = new ACHTransRecord;
$myachtransrecord3 -> SSS = "TST"; //T$$ assigned
$myachtransrecord3 -> LocID = "9505"; //T$$ assigned
$myachtransrecord3 -> FrontEndTrace = "W1000123451"; //impt!!!Needs a unique value each record per LocID
$myachtransrecord3 -> OriginatorName = "MYCOMPANY";
$myachtransrecord3 -> TransactionCode = "PPD"; //PPD, CCD, WEB, TEL, RCK, etc.
$myachtransrecord3 -> CustTransType = "C"; //D for debit, C for credit
$myachtransrecord3 -> CustomerID = "CustID12235";
$myachtransrecord3 -> CustomerName = "SMITH, JOE";
$myachtransrecord3 -> CustomerRoutingNo = "987654320";
$myachtransrecord3 -> CustomerAcctNo = "5500038883005";
$myachtransrecord3 -> CustomerAcctType = "C"; //C forchecking, S for Savings
$myachtransrecord3 -> TransAmount = 350.25;
$myachtransrecord3 -> CheckOrCustID = "9166388811";
$myachtransrecord3 -> CheckOrTransDate = '2010-03-16'; //include leading zero for mm and dd e.g. 01 for Jan
$myachtransrecord3 -> EffectiveDate = '2010-03-16'; //include leading zero for mm and dd e.g. 01 for Jan
$myachtransrecord3 -> Memo = "Payment";
$myachtransrecord3 -> OpCode = "S"; //S for Single, R for Recurring
$myachtransrecord3 -> AccountSet = "1"; //T$$ assigned

```

```

//ACHFile
$myachfile = new ACHFile;
$myachfile -> SSS = "TST";
$myachfile -> LocID = "9505";
$myachfile -> ACHFileName = ""; //leave blank to automatically name the file
$myachfile -> TotalNumRecords = 3;
$myachfile -> TotalDebitRecords = 2;
$myachfile -> TotalDebitAmount = 600.50;
$myachfile -> TotalCreditRecords = 1;
$myachfile -> TotalCreditAmount = 350.25;
//$myachfile -> ACHRecords = array($myachtransrecord1,$myachtransrecord2,$myachtransrecord3); //OR
$myachfile -> ACHRecords[] = $myachtransrecord1;
$myachfile -> ACHRecords[] = $myachtransrecord2;
$myachfile -> ACHRecords[] = $myachtransrecord3;

```

```

//SOAP Call - test server
$myclient = new SoapClient("http://tstsvr.achworks.com/dnet/achws.asmx?WSDL");
$myresult = $myclient->SendACHTransBatch(array("InpCompanyInfo"=>$mycompanyinfo,"InpACHFile"=>$myachfile))-
    >SendACHTransBatchResult;

```

```

//print status and details

```

```
print($myresult->Status . " , " . $myresult->Details . "<br>");
```

```
//print errors if there is any  
foreach ($myresult->Errors->string as $myerror) {  
print($myerror . "<br>");  
}  
?>
```

getachreturns.php

```
<?php  
//ACHWORKS-SOAP Ver3.0 GetACHReturns (gets unretrieved ach return and settlement records)  
//3.25.2010 - rico pamplona, rpamplonaATachworksDOTcom  
//company info  
class CompanyInfo {  
    public $SSS;  
    public $LocID;  
    public $Company;  
    public $CompanyKey;  
}  
  
//CompanyInfo  
$mycompanyinfo = new CompanyInfo;  
$mycompanyinfo -> SSS = "TST";  
$mycompanyinfo -> LocID = "9505";  
$mycompanyinfo -> Company = "THAT COMPANY";  
$mycompanyinfo -> CompanyKey = "RICO";  
  
//SOAP call - test server  
$myclient = new SoapClient("http://tstsvr.achworks.com/dnet/achws.asmx?WSDL");  
$myresult = $myclient->GetACHReturns(array("InpCompanyInfo"=>$mycompanyinfo))->GetACHReturnsResult;  
  
//print status and details  
print($myresult->Status . " , " . $myresult->Details . "<br>");  
  
//print ACHReturnRecords if there is any  
foreach ($myresult->ACHReturnRecords->ACHReturnRecord as $myACHReturnRecord) {  
print("FrontEndTrace:" . $myACHReturnRecord->FrontEndTrace . " , EffectiveDate:" . $myACHReturnRecord->EffectiveDate . " ,  
    Name:" . $myACHReturnRecord->CustomerName . " , Amount:" . $myACHReturnRecord->TransAmount . " ,  
    ResponseCode:" . $myACHReturnRecord->ResponseCode . " , ActionDetail:" . $myACHReturnRecord->ActionDetail .  
    "<br>");  
}  
?>
```

getachreturnshist.php

```
<?php  
//ACHWORKS-SOAP Ver3.0 GetACHReturnsHist (gets ach return and settlement records at specified date range)  
//3.25.2010 - rico pamplona, rpamplonaATachworksDOTcom  
//company info  
class CompanyInfo {  
    public $SSS;
```



```

    public $LocID;
    public $Company;
    public $CompanyKey;
}

//CompanyInfo
$mycompanyinfo = new CompanyInfo;
$mycompanyinfo -> SSS = "TST";
$mycompanyinfo -> LocID = "9505";
$mycompanyinfo -> Company = "THAT COMPANY";
$mycompanyinfo -> CompanyKey = "RICO";

$myDateFrom = '2010-01-01'; //include leading zero for mm and dd e.g. 01 for Jan
$myDateTo = '2010-03-20'; //include leading zero for mm and dd e.g. 01 for Jan

//SOAP call - test server
$myclient = new SoapClient("http://tstsvr.achworks.com/dnet/achws.asmx?WSDL");
$myresult = $myclient->GetACHReturnsHist(array("InpCompanyInfo"=>$mycompanyinfo, "ReturnDateFrom"=>$myDateFrom,
        "ReturnDateTo"=>$myDateTo))->GetACHReturnsHistResult;

//print status and details
print($myresult->Status . " , " . $myresult->Details . "<br>");

//print ACHReturnRecords if there is any
foreach ($myresult->ACHReturnRecords->ACHReturnRecord as $myACHReturnRecord) {
print("FrontEndTrace:" . $myACHReturnRecord->FrontEndTrace . " , EffectiveDate:" . $myACHReturnRecord->EffectiveDate . " ,
        Name:" . $myACHReturnRecord->CustomerName . " , Amount:" . $myACHReturnRecord->TransAmount . " ,
        ResponseCode:" . $myACHReturnRecord->ResponseCode . " , ActionDetail:" . $myACHReturnRecord->ActionDetail .
        "<br>");
}
?>

```

getresultfile.php

```

<?php
//ACHWORKS-SOAP Ver3.0 GetResultFile (gets send transaction results/responses including errors at specified date range)
//3.25.2010 - rico pamplona, rpamplonaATachworksDOTcom
//company info
class CompanyInfo {
    public $SSS;
    public $LocID;
    public $Company;
    public $CompanyKey;
}

//CompanyInfo
$mycompanyinfo = new CompanyInfo;
$mycompanyinfo -> SSS = "TST";
$mycompanyinfo -> LocID = "9505";
$mycompanyinfo -> Company = "THAT COMPANY";

```


A2.2 Visual Basic (VB.NET/ASP.NET)

connectioncheck

```
Sub Button1Click(sender As Object, e As EventArgs)
    'named nsACHWorksWS as the namespace when adding web reference
    Dim myACHWS As nsACHWorksWS.ACHWorksWS
    Dim myCompanyInfo As nsACHWorksWS.CompanyInfo

    'instances
    myACHWS = New nsACHWorksWS.ACHWorksWS
    myCompanyInfo = New nsACHWorksWS.CompanyInfo

    'CompanyInfo
    myCompanyInfo.SSS="TST"
    myCompanyInfo.LocID="9505"
    myCompanyInfo.Company="THAT COMPANY"
    myCompanyInfo.CompanyKey="RICO"

    'call ConnectionCheck and assign result to textBox1
    textBox1.Text=myACHWS.ConnectionCheck(myCompanyInfo)
End Sub
```

sendachtrans

```
Sub Button2Click(sender As Object, e As EventArgs)
    'named nsACHWorksWS as the namespace when adding web reference
    Dim myACHWS As nsACHWorksWS.ACHWorksWS
    Dim myCompanyInfo As nsACHWorksWS.CompanyInfo
    Dim myACHTransRecord As nsACHWorksWS.ACHTransRecord
    Dim myACHTransResult as nsACHWorksWS.TransResult

    'instances
    myACHWS = New nsACHWorksWS.ACHWorksWS
    myCompanyInfo = New nsACHWorksWS.CompanyInfo
    myACHTransRecord = New nsACHWorksWS.ACHTransRecord
    myACHTransResult = New nsACHWorksWS.TransResult

    'CompanyInfo
    myCompanyInfo.SSS="TST"
    myCompanyInfo.LocID="9505"
    myCompanyInfo.Company="THAT COMPANY"
    myCompanyInfo.CompanyKey="RICO"

    'ACHTransRecord
    myACHTransRecord.SSS="TST" 'T$$ assigned
    myACHTransRecord.LocID="9505" 'T$$ assigned
    myACHTransRecord.FrontEndTrace="VVB-0003" 'impt!!! needs a unique value each record per LocID
    myACHTransRecord.OriginatorName="MYCOMPANY"
```

```

myACHTransRecord.TransactionCode="PPD" 'PPD,CCD,WEB,TEL,RCK, etc
myACHTransRecord.CustTransType="D" 'D for debit, C for credit
myACHTransRecord.CustomerID="CustID0001"
myACHTransRecord.CustomerName="DOE, JOHN" 'Lastname, FirstName - all caps
myACHTransRecord.CustomerRoutingNo="987654320" '9-digit Bank Routing No
myACHTransRecord.CustomerAcctNo="034332222" 'Bank Account No
myACHTransRecord.CustomerAcctType="C" 'C for checking, S for savings
myACHTransRecord.TransAmount=100.75
myACHTransRecord.CheckOrCustID="9166388811"
myACHTransRecord.CheckOrTransDate="2010-03-24"
myACHTransRecord.EffectiveDate="2010-03-24"
myACHTransRecord.Memo="FirstPay"
myACHTransRecord.OpCode="S" 'S for single, R for recurring
myACHTransRecord.AccountSet="1" 'T$$ assigned

```

```
'call SendACHTrans method
```

```
myACHTransResult=myACHWS.SendACHTrans(myCompanyInfo,myACHTransRecord)
```

```
'assign portion of result to textBox1 (e.g. Status and Details)
```

```
textBox1.Text=myACHTransResult.Status + ", " + myACHTransResult.Details
```

```
'print errors to listBox if there is any
```

```
listBox1.Items.Clear
```

```
If myACHTransResult.TotalNumErrors>0 Then
```

```
For i=0 To myACHTransResult.TotalNumErrors-1
```

```
listBox1.Items.Add(myACHTransResult.Errors(i).ToString)
```

```
Next i
```

```
End If
```

```
End Sub
```

sendachtransbatch

```
Sub Button3Click(sender As Object, e As EventArgs)
```

```
'named nsACHWorksWS as the namespace when adding web reference
```

```
Dim myACHWS As nsACHWorksWS.ACHWorksWS
```

```
Dim myCompanyInfo As nsACHWorksWS.CompanyInfo
```

```
Dim myACHTransRecord1,myACHTransRecord2,myACHTransRecord3 As nsACHWorksWS.ACHTransRecord
```

```
Dim myACHFile As nsACHWorksWS.ACHFile
```

```
Dim myACHTransResult As nsACHWorksWS.TransResult
```

```
'instances
```

```
myACHWS = New nsACHWorksWS.ACHWorksWS
```

```
myCompanyInfo = New nsACHWorksWS.CompanyInfo
```

```
myACHTransRecord1 = New nsACHWorksWS.ACHTransRecord
```

```
myACHTransRecord2 = New nsACHWorksWS.ACHTransRecord
```

```
myACHTransRecord3 = New nsACHWorksWS.ACHTransRecord
```

```
myACHFile = new nsACHWorksWS.ACHFile
```

```
myACHTransResult = New nsACHWorksWS.TransResult
```

'CompanyInfo

```
myCompanyInfo.SSS="TST"  
myCompanyInfo.LocID="9505"  
myCompanyInfo.Company="THAT COMPANY"  
myCompanyInfo.CompanyKey="RICO"
```

'ACHFile

```
myACHFile.SSS="TST"  
myACHFile.LocID="9505"  
myACHFile.ACHFileName="" 'leave blank to automatically name the file  
myACHFile.TotalNumRecords=3  
myACHFile.TotalDebitRecords=2  
myACHFile.TotalDebitAmount=600.50  
myACHFile.TotalCreditRecords=1  
myACHFile.TotalCreditAmount=350.25  
myACHFile.ACHRecords = new nsACHWorksWS.ACHTransRecord(2){} '(2) is 3-1
```

'assign records to file

'ACHTransRecord (1, 2 & 3)

'1st record

```
myACHTransRecord1.SSS="TST" 'T$$ assigned  
myACHTransRecord1.LocID="9505" 'T$$ assigned  
myACHTransRecord1.FrontEndTrace="VVB-0003" 'impt!!! needs a unique value each record per LocID  
myACHTransRecord1.OriginatorName="MYCOMPANY"  
myACHTransRecord1.TransactionCode="PPD" 'PPD,CCD,WEB,TEL,RCK, etc  
myACHTransRecord1.CustTransType="D" 'D for debit, C for credit  
myACHTransRecord1.CustomerID="CustID0001"  
myACHTransRecord1.CustomerName="DOE, JOHN" 'Lastname, FirstName - all caps  
myACHTransRecord1.CustomerRoutingNo="987654320" '9-digit Bank Routing No  
myACHTransRecord1.CustomerAcctNo="034332222" 'Bank Account No  
myACHTransRecord1.CustomerAcctType="C" 'C for checking, S for savings  
myACHTransRecord1.TransAmount=100.25  
myACHTransRecord1.CheckOrCustID="9166388811"  
myACHTransRecord1.CheckOrTransDate="2010-03-24"  
myACHTransRecord1.EffectiveDate="2010-03-24"  
myACHTransRecord1.Memo="FirstPay"  
myACHTransRecord1.OpCode="S" 'S for single, R for recurring  
myACHTransRecord1.AccountSet="1" 'T$$ assigned  
myACHFile.ACHRecords(0)=new nsACHWorksWS.ACHTransRecord 'instance  
myACHFile.ACHRecords(0)=myACHTransRecord1 'instance
```

'2nd record

```
myACHTransRecord2.SSS="TST" 'T$$ assigned  
myACHTransRecord2.LocID="9505" 'T$$ assigned  
myACHTransRecord2.FrontEndTrace="VVB-0004" 'impt!!! needs a unique value each record per LocID  
myACHTransRecord2.OriginatorName="MYCOMPANY"  
myACHTransRecord2.TransactionCode="PPD" 'PPD,CCD,WEB,TEL,RCK, etc  
myACHTransRecord2.CustTransType="D" 'D for debit, C for credit
```

```

myACHTransRecord2.CustomerID="CustID0002"
myACHTransRecord2.CustomerName="DOE, JANE" 'Lastname, FirstName - all caps
myACHTransRecord2.CustomerRoutingNo="987654320" '9-digit Bank Routing No
myACHTransRecord2.CustomerAcctNo="111034332222" 'Bank Account No
myACHTransRecord2.CustomerAcctType="C" 'C for checking, S for savings
myACHTransRecord2.TransAmount=500.25
myACHTransRecord2.CheckOrCustID="9166388811"
myACHTransRecord2.CheckOrTransDate="2010-03-24"
myACHTransRecord2.EffectiveDate="2010-03-24"
myACHTransRecord2.Memo="FirstPay"
myACHTransRecord2.OpCode="S" 'S for single, R for recurring
myACHTransRecord2.AccountSet="1" 'T$$ assigned
myACHFile.ACHRecords(1)=new nsACHWorksWS.ACHTransRecord 'instance
myACHFile.ACHRecords(1)=myACHTransRecord2

```

'3rd record

```

myACHTransRecord3.SSS="TST" 'T$$ assigned
myACHTransRecord3.LocID="9505" 'T$$ assigned
myACHTransRecord3.FrontEndTrace="WVB-0005" 'impt!!! needs a unique value each record per LocID
myACHTransRecord3.OriginatorName="MYCOMPANY"
myACHTransRecord3.TransactionCode="PPD" 'PPD,CCD,WEB,TEL,RCK, etc
myACHTransRecord3.CustTransType="C" 'credit
myACHTransRecord3.CustomerID="CustID0003"
myACHTransRecord3.CustomerName="SMITH, JOE" 'Lastname, FirstName - all caps
myACHTransRecord3.CustomerRoutingNo="987654320" '9-digit Bank Routing No
myACHTransRecord3.CustomerAcctNo="111034332222" 'Bank Account No
myACHTransRecord3.CustomerAcctType="C" 'C for checking, S for savings
myACHTransRecord3.TransAmount=350.25
myACHTransRecord3.CheckOrCustID="9166388811"
myACHTransRecord3.CheckOrTransDate="2010-03-24"
myACHTransRecord3.EffectiveDate="2010-03-24"
myACHTransRecord3.Memo="FirstPay"
myACHTransRecord3.OpCode="S" 'S for single, R for recurring
myACHTransRecord3.AccountSet="1" 'T$$ assigned
myACHFile.ACHRecords(2)=new nsACHWorksWS.ACHTransRecord 'instance
myACHFile.ACHRecords(2)=myACHTransRecord3

```

'call SendACHTransBatch method

```

myACHTransResult=myACHWS.SendACHTransBatch(myCompanyInfo,myACHFile)

```

'assign portion of result to textBox1 (e.g. Status and Details)

```

textBox1.Text=myACHTransResult.Status + ", " + myACHTransResult.Details

```

'print errors to listBox if there is any

```

listBox1.Items.Clear

```

```

If myACHTransResult.TotalNumErrors>0 Then

```

```

    For i=0 To myACHTransResult.TotalNumErrors-1

```

```

        listBox1.Items.Add(myACHTransResult.Errors(i).ToString)

```

```

    Next i

```

```
End If
End Sub
```

getachreturns

```
Sub Button4Click(sender As Object, e As EventArgs)
    Dim myACHWS As nsACHWorksWS.ACHWorksWS
    Dim myCompanyInfo As nsACHWorksWS.CompanyInfo
    Dim myACHReturns As nsACHWorksWS.ACHReturns

    'instances
    myACHWS = New nsACHWorksWS.ACHWorksWS
    myCompanyInfo = New nsACHWorksWS.CompanyInfo
    myACHReturns = New nsACHWorksWS.ACHReturns

    'CompanyInfo
    myCompanyInfo.SSS="TST"
    myCompanyInfo.LocID="9505"
    myCompanyInfo.Company="THAT COMPANY"
    myCompanyInfo.CompanyKey="RICO"

    'call GetACHReturns method
    myACHReturns = myACHWS.GetACHReturns(myCompanyInfo)
    textBox1.Text=myACHReturns.Status + ", " + myACHReturns.Details

    'list returns if there are any
    listBox1.Items.Clear
    If myACHReturns.TotalNumRecords>0 Then
        For i=0 To myACHReturns.TotalNumRecords-1
            listBox1.Items.Add("FrontEndTrace:" + myACHReturns.ACHReturnRecords(i).FrontEndTrace + _
                ", Date:" + myACHReturns.ACHReturnRecords(i).EffectiveDate.ToString + _
                ", Name:" + myACHReturns.ACHReturnRecords(i).CustomerName + _
                ", Amount:" + myACHReturns.ACHReturnRecords(i).TransAmount.ToString + _
                ", ResponseCode:" + myACHReturns.ACHReturnRecords(i).ResponseCode + _
                ", ActionDetail:" + myACHReturns.ACHReturnRecords(i).ActionDetail)
        Next i
    End If
End Sub
```

getachreturnshist

```
Sub Button5Click(sender As Object, e As EventArgs)
    Dim myACHWS As nsACHWorksWS.ACHWorksWS
    Dim myCompanyInfo As nsACHWorksWS.CompanyInfo
    Dim myACHReturns As nsACHWorksWS.ACHReturns
    Dim myDateFrom,myDateTo as Date
```



```
'instances
```

```
myACHWS = New nsACHWorksWS.ACHWorksWS  
myCompanyInfo = New nsACHWorksWS.CompanyInfo  
myACHReturns = New nsACHWorksWS.ACHReturns
```

```
'CompanyInfo
```

```
myCompanyInfo.SSS="TST"  
myCompanyInfo.LocID="9505"  
myCompanyInfo.Company="THAT COMPANY"  
myCompanyInfo.CompanyKey="RICO"
```

```
'date
```

```
myDateFrom = "2010-01-01"  
myDateTo = "2010-03-20"
```

```
'call GetACHReturnsHist method
```

```
myACHReturns = myACHWS.GetACHReturnsHist(myCompanyInfo,myDateFrom,myDateTo)  
textBox1.Text=myACHReturns.Status + ", " + myACHReturns.Details
```

```
'list returns if there are any
```

```
listBox1.Items.Clear
```

```
If myACHReturns.TotalNumRecords>0 Then
```

```
For i=0 To myACHReturns.TotalNumRecords-1
```

```
listBox1.Items.Add("FrontEndTrace:" + myACHReturns.ACHReturnRecords(i).FrontEndTrace + _  
", Date:" + myACHReturns.ACHReturnRecords(i).EffectiveDate.ToString + _  
", Name:" + myACHReturns.ACHReturnRecords(i).CustomerName + _  
", Amount:" + myACHReturns.ACHReturnRecords(i).TransAmount.ToString + _  
", ResponseCode:" + myACHReturns.ACHReturnRecords(i).ResponseCode + _  
", ActionDetail:" + myACHReturns.ACHReturnRecords(i).ActionDetail)
```

```
Next i
```

```
End If
```

```
End Sub
```

getresultfile

```
Sub Button6Click(sender As Object, e As EventArgs)
```

```
Dim myACHWS As nsACHWorksWS.ACHWorksWS
```

```
Dim myCompanyInfo As nsACHWorksWS.CompanyInfo
```

```
Dim myResultFile As nsACHWorksWS.ResultFile
```

```
Dim myDateFrom,myDateTo as Date
```

```
'instances
```

```
myACHWS = New nsACHWorksWS.ACHWorksWS  
myCompanyInfo = New nsACHWorksWS.CompanyInfo  
myResultFile = New nsACHWorksWS.ResultFile
```

```

'CompanyInfo
myCompanyInfo.SSS="TST"
myCompanyInfo.LocID="9505"
myCompanyInfo.Company="THAT COMPANY"
myCompanyInfo.CompanyKey="RICO"

'date
myDateFrom = "2010-03-25"
myDateTo = "2010-03-26"

'call GetResultFile method
myResultFile = myACHWS.GetResultFile(myCompanyInfo,myDateFrom,myDateTo)

'assign status/details to textbox
textBox1.Text=myResultFile.Status + ", " + myResultFile.Details

'list past connection results to listBox1
listBox1.Items.Clear
If myResultFile.TotalResultRecords>0 Then
  For i=0 To myResultFile.TotalResultRecords-1
    listBox1.Items.Add("DateTime:" + myResultFile.TransResults(i).CallDateTime.ToString + _
      ", Method:" + myResultFile.TransResults(i).CallMethod + _
      ", Status:" + myResultFile.TransResults(i).Status + _
      ", FileName:" + myResultFile.TransResults(i).FileName)
    For j=0 To myResultFile.TransResults(i).TotalNumErrors-1
      listBox1.Items.Add(">" + myResultFile.TransResults(i).Errors(j).ToString)
    Next j
  Next i
End If
End Sub

```

geterrorfile

```

Sub Button7Click(sender As Object, e As EventArgs)
  Dim myACHWS As nsACHWorksWS.ACHWorksWS
  Dim myCompanyInfo As nsACHWorksWS.CompanyInfo
  Dim myErrorFile As nsACHWorksWS.ErrorFile
  Dim myDateFrom,myDateTo as Date

  'instances
  myACHWS = New nsACHWorksWS.ACHWorksWS
  myCompanyInfo = New nsACHWorksWS.CompanyInfo
  myErrorFile = New nsACHWorksWS.ErrorFile

```

```

'CompanyInfo
myCompanyInfo.SSS="TST"
myCompanyInfo.LocID="9505"
myCompanyInfo.Company="THAT COMPANY"
myCompanyInfo.CompanyKey="RICO"

'date
myDateFrom = "2010-03-25"
myDateTo = "2010-03-26"

'call GetErrorFile method
myErrorFile = myACHWS.GetErrorFile(myCompanyInfo,myDateFrom,myDateTo)

'assign status/details to textbox
textBox1.Text=myErrorFile.Status + ", " + myErrorFile.Details

'list past connection errors to listBox1
listBox1.Items.Clear
If myErrorFile.TotalErrorRecords>0 Then
  For i=0 To myErrorFile.TotalErrorRecords-1
    listBox1.Items.Add("DateTime:" + myErrorFile.ErrorRecords(i).CallDateTime.ToString + _
                      ", Method:" + myErrorFile.ErrorRecords(i).CallMethod + _
                      ", FileName:" + myErrorFile.ErrorRecords(i).FileName + _
                      ", No of Errors:" + myErrorFile.ErrorRecords(i).TotalNumErrors.ToString)
    For j=0 To myErrorFile.ErrorRecords(i).TotalNumErrors-1
      listBox1.Items.Add(">" + myErrorFile.ErrorRecords(i).Errors(j).ToString)
    Next j
  Next i
End If
End Sub

```

A2.3 C# (.NET)

connectioncheck

```
void Button1Click(object sender, EventArgs e)
{
    //named nsACHWorksWS as the namespace when adding web reference
    //instances
    nsACHWorksWS.ACHWorksWS myACHWS = new nsACHWorksWS.ACHWorksWS();
    nsACHWorksWS.CompanyInfo myCompanyInfo = new nsACHWorksWS.CompanyInfo();

    //company info
    myCompanyInfo.SSS="TST";
    myCompanyInfo.LocID="9505";
    myCompanyInfo.Company="THAT COMPANY";
    myCompanyInfo.CompanyKey="RICO";

    //call ConnectionCheck method
    textBox1.Text=myACHWS.ConnectionCheck(myCompanyInfo);
}
```

sendachtrans

```
void Button2Click(object sender, EventArgs e)
{
    //instances
    nsACHWorksWS.ACHWorksWS myACHWS = new nsACHWorksWS.ACHWorksWS();
    nsACHWorksWS.CompanyInfo myCompanyInfo = new nsACHWorksWS.CompanyInfo();
    nsACHWorksWS.ACHTransRecord myACHTransRecord = new nsACHWorksWS.ACHTransRecord();
    nsACHWorksWS.TransResult myACHTransResult = new nsACHWorksWS.TransResult();

    //company info
    myCompanyInfo.SSS="TST";
    myCompanyInfo.LocID="9505";
    myCompanyInfo.Company="THAT COMPANY";
    myCompanyInfo.CompanyKey="RICO";

    //ACHTransRecord
    myACHTransRecord.SSS="TST"; //T$$ assigned
    myACHTransRecord.LocID="9505"; //T$$ assigned
    myACHTransRecord.FrontEndTrace="CC#-0003"; //impt! needs a unique value each record per LocID
    myACHTransRecord.OriginatorName="MYCOMPANY";
    myACHTransRecord.TransactionCode="PPD"; //PPD,CCD,WEB,TEL,RCK, etc.
    myACHTransRecord.CustTransType="D"; //D for debit, C for credit
    myACHTransRecord.CustomerID="CustID0001";
}
```

```

myACHTransRecord.CustomerName="DOE, JOHN"; //Lastname, Firstname - all caps
myACHTransRecord.CustomerRoutingNo="987654320"; //9-digit Bank Routing No
myACHTransRecord.CustomerAcctNo="0002311331555"; //Bank Account No
myACHTransRecord.CustomerAcctType="C"; //C for checking, S for savings
myACHTransRecord.TransAmount=100.75;
myACHTransRecord.CheckOrCustID="9166388811";
myACHTransRecord.CheckOrTransDate=Convert.ToDateTime("2010-03-24");
myACHTransRecord.EffectiveDate=Convert.ToDateTime("2010-03-24");
myACHTransRecord.Memo="FirstPay";
myACHTransRecord.OpCode="S"; //S for single, R for recurring
myACHTransRecord.AccountSet="1"; //T$$ assigned

```

```

//call SendACHTrans method
myACHTransResult=myACHWS.SendACHTrans(myCompanyInfo,myACHTransRecord);

```

```

//assign portion of result to textbox1 (e.g. status and details)
textBox1.Text=myACHTransResult.Status + ", " + myACHTransResult.Details;

```

```

//print errors to listBox if there is any
listBox1.Items.Clear();
if (myACHTransResult.TotalNumErrors>0)
{
    for (int i=0; i<myACHTransResult.TotalNumErrors; i++)
        listBox1.Items.Add(myACHTransResult.Errors[i]);
}
}

```

sendachtransbatch

```

void Button3Click(object sender, EventArgs e)
{

```

```

    //instances
    nsACHWorksWS.ACHWorksWS myACHWS = new nsACHWorksWS.ACHWorksWS();
    nsACHWorksWS.CompanyInfo myCompanyInfo = new nsACHWorksWS.CompanyInfo();
    nsACHWorksWS.ACHTransRecord myACHTransRecord1 = new nsACHWorksWS.ACHTransRecord();
    nsACHWorksWS.ACHTransRecord myACHTransRecord2 = new nsACHWorksWS.ACHTransRecord();
    nsACHWorksWS.ACHTransRecord myACHTransRecord3 = new nsACHWorksWS.ACHTransRecord();
    nsACHWorksWS.ACHFile myACHFile = new nsACHWorksWS.ACHFile();
    nsACHWorksWS.TransResult myACHTransResult = new nsACHWorksWS.TransResult();

```

```

    //company info
    myCompanyInfo.SSS="TST";
    myCompanyInfo.LocID="9505";
    myCompanyInfo.Company="THAT COMPANY";
    myCompanyInfo.CompanyKey="RICO";

```

```

    //ACHFile
    myACHFile.SSS="TST";

```

```
myACHFile.LocID="9505";
myACHFile.ACHFileName=""; //leave blank to automatically name the file
myACHFile.TotalNumRecords=3;
myACHFile.TotalDebitRecords=2;
myACHFile.TotalDebitAmount=600.50;
myACHFile.TotalCreditRecords=1;
myACHFile.TotalCreditAmount=350.25;
myACHFile.ACHRecords=new nsACHWorksWS.ACHTransRecord[3];
```

```
//ACHTransRecord(s)
//Record 1
```

```
myACHTransRecord1.SSS="TST"; //T$$ assigned
myACHTransRecord1.LocID="9505"; //T$$ assigned
myACHTransRecord1.FrontEndTrace="CC#-00031"; //impt! needs a unique value each record per LocID
myACHTransRecord1.OriginatorName="MYCOMPANY";
myACHTransRecord1.TransactionCode="PPD"; //PPD,CCD,WEB,TEL,RCK, etc.
myACHTransRecord1.CustTransType="D"; //D for debit, C for credit
myACHTransRecord1.CustomerID="CustID0001";
myACHTransRecord1.CustomerName="DOE, JOHN"; //Lastname, Firstname - all caps
myACHTransRecord1.CustomerRoutingNo="987654320"; //9-digit Bank Routing No
myACHTransRecord1.CustomerAcctNo="0002311331555"; //Bank Account No
myACHTransRecord1.CustomerAcctType="C"; //C for checking, S for savings
myACHTransRecord1.TransAmount=100.25;
myACHTransRecord1.CheckOrCustID="9166388811";
myACHTransRecord1.CheckOrTransDate=Convert.ToDateTime("2010-03-24");
myACHTransRecord1.EffectiveDate=Convert.ToDateTime("2010-03-24");
myACHTransRecord1.Memo="FirstPay";
myACHTransRecord1.OpCode="S"; //S for single, R for recurring
myACHTransRecord1.AccountSet="1"; //T$$ assigned
myACHFile.ACHRecords[0]=new nsACHWorksWS.ACHTransRecord();
myACHFile.ACHRecords[0]=myACHTransRecord1;
```

```
//Record 2
```

```
myACHTransRecord2.SSS="TST"; //T$$ assigned
myACHTransRecord2.LocID="9505"; //T$$ assigned
myACHTransRecord2.FrontEndTrace="CC#-00032"; //impt! needs a unique value each record per LocID
myACHTransRecord2.OriginatorName="MYCOMPANY";
myACHTransRecord2.TransactionCode="PPD"; //PPD,CCD,WEB,TEL,RCK, etc.
myACHTransRecord2.CustTransType="D"; //D for debit, C for credit
myACHTransRecord2.CustomerID="CustID0002";
myACHTransRecord2.CustomerName="DOE, JANE"; //Lastname, Firstname - all caps
myACHTransRecord2.CustomerRoutingNo="987654320"; //9-digit Bank Routing No
myACHTransRecord2.CustomerAcctNo="400231133002"; //Bank Account No
myACHTransRecord2.CustomerAcctType="C"; //C for checking, S for savings
myACHTransRecord2.TransAmount=500.25;
myACHTransRecord2.CheckOrCustID="9166388811";
myACHTransRecord2.CheckOrTransDate=Convert.ToDateTime("2010-03-24");
myACHTransRecord2.EffectiveDate=Convert.ToDateTime("2010-03-24");
myACHTransRecord2.Memo="FirstPay";
myACHTransRecord2.OpCode="S"; //S for single, R for recurring
myACHTransRecord2.AccountSet="1"; //T$$ assigned
myACHFile.ACHRecords[1]=new nsACHWorksWS.ACHTransRecord();
```

```
myACHFile.ACHRecords[1]=myACHTransRecord2;
```

```
//Record 3
```

```
myACHTransRecord3.SSS="TST"; //T$$ assigned  
myACHTransRecord3.LocID="9505"; //T$$ assigned  
myACHTransRecord3.FrontEndTrace="CC#-00033"; //impt! needs a unique value each record per LocID  
myACHTransRecord3.OriginatorName="MYCOMPANY";  
myACHTransRecord3.TransactionCode="PPD"; //PPD,CCD,WEB,TEL,RCK, etc.  
myACHTransRecord3.CustTransType="C"; //D for debit, C for credit  
myACHTransRecord3.CustomerID="CustID0003";  
myACHTransRecord3.CustomerName="SMITH, JOE"; //Lastname, Firstname - all caps  
myACHTransRecord3.CustomerRoutingNo="987654320"; //9-digit Bank Routing No  
myACHTransRecord3.CustomerAcctNo="90200231552"; //Bank Account No  
myACHTransRecord3.CustomerAcctType="C"; //C for checking, S for savings  
myACHTransRecord3.TransAmount=350.25;  
myACHTransRecord3.CheckOrCustID="9166388811";  
myACHTransRecord3.CheckOrTransDate=Convert.ToDateTime("2010-03-24");  
myACHTransRecord3.EffectiveDate=Convert.ToDateTime("2010-03-24");  
myACHTransRecord3.Memo="FirstPay";  
myACHTransRecord3.OpCode="S"; //S for single, R for recurring  
myACHTransRecord3.AccountSet="1"; //T$$ assigned  
myACHFile.ACHRecords[2]=new nsACHWorksWS.ACHTransRecord();  
myACHFile.ACHRecords[2]=myACHTransRecord3;
```

```
//call SendACHTransBatch method
```

```
myACHTransResult=myACHWS.SendACHTransBatch(myCompanyInfo,myACHFile);
```

```
//assign portion of result to textbox1 (e.g. status and details)
```

```
textBox1.Text=myACHTransResult.Status + ", " + myACHTransResult.Details;
```

```
//print errors to listBox if there is any
```

```
listBox1.Items.Clear();
```

```
if (myACHTransResult.TotalNumErrors>0)
```

```
{
```

```
    for (int i=0; i<myACHTransResult.TotalNumErrors; i++)
```

```
        listBox1.Items.Add(myACHTransResult.Errors[i]);
```

```
    }
```

```
}
```

getachreturns

```
void Button4Click(object sender, EventArgs e)
```

```
{
```

```
    //instances
```

```
    nsACHWorksWS.ACHWorksWS myACHWS = new nsACHWorksWS.ACHWorksWS();
```

```
    nsACHWorksWS.CompanyInfo myCompanyInfo = new nsACHWorksWS.CompanyInfo();
```

```
    nsACHWorksWS.ACHReturns myACHReturns = new nsACHWorksWS.ACHReturns();
```

```

//company info
myCompanyInfo.SSS="TST";
myCompanyInfo.LocID="9505";
myCompanyInfo.Company="THAT COMPANY";
myCompanyInfo.CompanyKey="RICO";

//call GetACHReturns method
myACHReturns=myACHWS.GetACHReturns(myCompanyInfo);
textBox1.Text=myACHReturns.Status + ", " + myACHReturns.Details;

//list returns if there is any
listBox1.Items.Clear();
if (myACHReturns.TotalNumRecords>0)
{
    for (int i=0; i<myACHReturns.TotalNumRecords; i++)
        listBox1.Items.Add("FrontEndTrace:" + myACHReturns.ACHReturnRecords[i].FrontEndTrace +
            ", EffectiveDate:" + myACHReturns.ACHReturnRecords[i].EffectiveDate.ToString() +
            ", Name:" + myACHReturns.ACHReturnRecords[i].CustomerName +
            ", Amount:" + myACHReturns.ACHReturnRecords[i].TransAmount.ToString() +
            ", ResponseCode:" + myACHReturns.ACHReturnRecords[i].ResponseCode +
            ", ActionDetail:" + myACHReturns.ACHReturnRecords[i].ActionDetail.Trim() +
            ", ActionDate:" + myACHReturns.ACHReturnRecords[i].ActionDate.ToString());
}
}

```

getachreturnshist

```

void Button5Click(object sender, EventArgs e)
{
    //instances
    nsACHWorksWS.ACHWorksWS myACHWS = new nsACHWorksWS.ACHWorksWS();
    nsACHWorksWS.CompanyInfo myCompanyInfo = new nsACHWorksWS.CompanyInfo();
    nsACHWorksWS.ACHReturns myACHReturns = new nsACHWorksWS.ACHReturns();
    DateTime myDateFrom, myDateTo;

    //company info
    myCompanyInfo.SSS="TST";
    myCompanyInfo.LocID="9505";
    myCompanyInfo.Company="THAT COMPANY";
    myCompanyInfo.CompanyKey="RICO";

    //date
    myDateFrom=Convert.ToDateTime("2010-01-01");
    myDateTo=Convert.ToDateTime("2010-03-20");
}

```



```

//call GetACHReturnsHist method
myACHReturns=myACHWS.GetACHReturnsHist(myCompanyInfo,myDateFrom,myDateTo);
textBox1.Text=myACHReturns.Status + ", " + myACHReturns.Details;

//list returns if there is any
listBox1.Items.Clear();
if (myACHReturns.TotalNumRecords>0)
{
    for (int i=0; i<myACHReturns.TotalNumRecords; i++)
        listBox1.Items.Add("FrontEndTrace:" + myACHReturns.ACHReturnRecords[i].FrontEndTrace +
            ", EffectiveDate:" + myACHReturns.ACHReturnRecords[i].EffectiveDate.ToString() +
            ", Name:" + myACHReturns.ACHReturnRecords[i].CustomerName +
            ", Amount:" + myACHReturns.ACHReturnRecords[i].TransAmount.ToString() +
            ", ResponseCode:" + myACHReturns.ACHReturnRecords[i].ResponseCode +
            ", ActionDetail:" + myACHReturns.ACHReturnRecords[i].ActionDetail +
            ", ActionDate:" + myACHReturns.ACHReturnRecords[i].ActionDate.ToString());
}
}

```

getresultfile

```

void Button6Click(object sender, EventArgs e)
{
    //instances
    nsACHWorksWS.ACHWorksWS myACHWS = new nsACHWorksWS.ACHWorksWS();
    nsACHWorksWS.CompanyInfo myCompanyInfo = new nsACHWorksWS.CompanyInfo();
    nsACHWorksWS.ResultFile myResultFile = new nsACHWorksWS.ResultFile();
    DateTime myDateFrom, myDateTo;

    //company info
    myCompanyInfo.SSS="TST";
    myCompanyInfo.LocID="9505";
    myCompanyInfo.Company="THAT COMPANY";
    myCompanyInfo.CompanyKey="RICO";

    //date
    myDateFrom=Convert.ToDateTime("2010-03-25");
    myDateTo=Convert.ToDateTime("2010-03-26");

    //call GetResultFile method
    myResultFile=myACHWS.GetResultFile(myCompanyInfo,myDateFrom,myDateTo);
    textBox1.Text=myResultFile.Status + ", " + myResultFile.Details;
}

```

```

//list past connection results
listBox1.Items.Clear();
if (myResultFile.TotalResultRecords>0)
{
    for (int i=0; i<myResultFile.TotalResultRecords; i++)
    {
        listBox1.Items.Add("DateTime:" + myResultFile.TransResults[i].CallDateTime.ToString() +
            ", Method:" + myResultFile.TransResults[i].CallMethod +
            ", Status:" + myResultFile.TransResults[i].Status +
            ", FileName:" + myResultFile.TransResults[i].FileName);
        for (int j=0; j<myResultFile.TransResults[i].TotalNumErrors; j++)
            listBox1.Items.Add(">" + myResultFile.TransResults[i].Errors[j]);
    }
}
}
}

```

geterrorfile

```

void Button7Click(object sender, EventArgs e)
{
    //instances
    nsACHWorksWS.ACHWorksWS myACHWS = new nsACHWorksWS.ACHWorksWS();
    nsACHWorksWS.CompanyInfo myCompanyInfo = new nsACHWorksWS.CompanyInfo();
    nsACHWorksWS.ErrorFile myErrorFile = new nsACHWorksWS.ErrorFile();
    DateTime myDateFrom, myDateTo;

    //company info
    myCompanyInfo.SSS="TST";
    myCompanyInfo.LocID="9505";
    myCompanyInfo.Company="THAT COMPANY";
    myCompanyInfo.CompanyKey="RICO";

    //date
    myDateFrom=Convert.ToDateTime("2010-03-25");
    myDateTo=Convert.ToDateTime("2010-03-26");

    //call GetErrorFile method
    myErrorFile=myACHWS.GetErrorFile(myCompanyInfo,myDateFrom,myDateTo);
    textBox1.Text=myErrorFile.Status + ", " + myErrorFile.Details;

    //list error records
    listBox1.Items.Clear();
    if (myErrorFile.TotalErrorRecords>0)
    {
        for (int i=0; i<myErrorFile.TotalErrorRecords; i++)
    }
}
}

```

```
{
  listBox1.Items.Add("DateTime:" + myErrorFile.ErrorRecords[i].CallDateTime.ToString() +
    ", Method:" + myErrorFile.ErrorRecords[i].CallMethod +
    ", FileName:" + myErrorFile.ErrorRecords[i].FileName +
    ", No of Errors:" + myErrorFile.ErrorRecords[i].TotalNumErrors.ToString());
  for (int j=0; j< myErrorFile.ErrorRecords[i].TotalNumErrors; j++)
    listBox1.Items.Add(">" + myErrorFile.ErrorRecords[i].Errors[j]);
}
}
```

A2.4 Java (1.5/1.6)

connectioncheck

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    try
    {
        //instances
        ACHWorksWS myACHWS = new ACHWorksWS();
        CompanyInfo myCompanyInfo = new CompanyInfo();

        //assign values for myCompanyInfo
        myCompanyInfo.sss="TST";
        myCompanyInfo.locID="9505";
        myCompanyInfo.company="THAT COMPANY";
        myCompanyInfo.companyKey="RICO";

        //put value of ConnectionCheck call to jTextField
        jTextField1.setText(myACHWS.getACHWorksWS SOAP().connectionCheck(myCompanyInfo));
    }
    catch (Exception e)
    {
        jTextField1.setText("Exception caught: " + e);
    }
}
```

sendachtrans

```
private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
    //need to 'import' the following:
    //import java.util.*;
    //import java.text.*;
    //import javax.xml.datatype.DatatypeConfigurationException;
    //import javax.xml.datatype.DatatypeFactory;
    //import javax.xml.datatype.XMLGregorianCalendar;
    //import javax.swing.DefaultListModel;

    try
    {
        //instances
        ACHWorksWS myACHWS = new ACHWorksWS();
        CompanyInfo myCompanyInfo = new CompanyInfo();
        ACHTransRecord myACHTransRecord = new ACHTransRecord();
        TransResult myACHTransResult = new TransResult();
        DatatypeFactory myDTF;

        //assign values for myCompanyInfo
        myCompanyInfo.sss="TST";
        myCompanyInfo.locID="9505";
        myCompanyInfo.company="THAT COMPANY";
        myCompanyInfo.companyKey="RICO";
    }
}
```

```

//ACHTransRecord
myACHTransRecord.sss="TST"; //T$$ assigned
myACHTransRecord.locID="9505"; //T$$ assigned
myACHTransRecord.frontEndTrace="JJ-0007"; //impt! needs a unique value for each record per locid
myACHTransRecord.originatorName="MYCOMPANY";
myACHTransRecord.transactionCode="PPD"; //PPD,CCD,WEB,TEL,RCK, etc.
myACHTransRecord.custTransType="D"; //D for debit, C for credit
myACHTransRecord.customerID="CUSTID123";
myACHTransRecord.customerName="DOE, JOHN"; //Lastname, Firstname - all caps
myACHTransRecord.customerRoutingNo="987654320"; //9-digit Bank Routing No
myACHTransRecord.customerAcctNo="00332358882"; //Bank Account No
myACHTransRecord.customerAcctType="C"; //C for checking, S for savings
myACHTransRecord.transAmount=100.75;
myACHTransRecord.checkOrCustID="9166388811";

try
{
    myDTF = DatatypeFactory.newInstance();
    myACHTransRecord.checkOrTransDate=myDTF.newXMLGregorianCalendar("2010-03-24T00:00:00");
    myACHTransRecord.effectiveDate=myDTF.newXMLGregorianCalendar("2010-03-24T00:00:00");
}
catch (DatatypeConfigurationException e)
{
    jTextField1.setText("Date exception caught: " + e);
}

myACHTransRecord.memo="FirstPay";
myACHTransRecord.opCode="S"; //S for single, R for recurring
myACHTransRecord.accountSet="1"; //T$$ assigned

//call SendACHTrans method
myACHTransResult = myACHWS.getACHWorksWSSoap().sendACHTrans(myCompanyInfo, myACHTransRecord);
jTextField1.setText(myACHTransResult.status + ", " + myACHTransResult.details);

//list errors if there are any
if (myACHTransResult.totalNumErrors>0)
{
    DefaultListModel myList = new DefaultListModel();

    jList1.setVisibleRowCount(myACHTransResult.totalNumErrors);
    for (int i=0; i<myACHTransResult.totalNumErrors; i++)
        myList.addElement(myACHTransResult.errors.string.get(i));

    jList1.setModel(myList);
}
}
catch (Exception e)
{
    jTextField1.setText("Exception caught: " + e);
}
}

```

sendachtransbatch

```
private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {
    //need to 'import' the following:
    //import java.util.*;
    //import java.text.*;
    //import javax.xml.datatype.DatatypeConfigurationException;
    //import javax.xml.datatype.DatatypeFactory;
    //import javax.xml.datatype.XMLGregorianCalendar;
    //import javax.swing.DefaultListModel;

    try
    {
        //instances
        ACHWorksWS myACHWS = new ACHWorksWS();
        CompanyInfo myCompanyInfo = new CompanyInfo();
        ACHTransRecord myACHTransRecord1 = new ACHTransRecord();
        ACHTransRecord myACHTransRecord2 = new ACHTransRecord();
        ACHTransRecord myACHTransRecord3 = new ACHTransRecord();
        ACHFile myACHFile = new ACHFile();
        TransResult myACHTransResult = new TransResult();
        DatatypeFactory myDTF;

        //assign values for myCompanyInfo
        myCompanyInfo.sss="TST";
        myCompanyInfo.locID="9505";
        myCompanyInfo.company="THAT COMPANY";
        myCompanyInfo.companyKey="RICO";

        //myACHFile
        myACHFile.sss="TST";
        myACHFile.locID="9505";
        myACHFile.achFileName=""; //leave blank to automatically name the file
        myACHFile.totalNumRecords=3;
        myACHFile.totalDebitRecords=2;
        myACHFile.totalDebitAmount=600.50;
        myACHFile.totalCreditRecords=1;
        myACHFile.totalCreditAmount=350.25;

        //initialize records
        myACHFile.achRecords = new ArrayOfACHTransRecord();

        //ACHTransRecord(s)
        //Record 1
        myACHTransRecord1.sss="TST"; //T$$ assigned
        myACHTransRecord1.locID="9505"; //T$$ assigned
        myACHTransRecord1.frontEndTrace="JJ-00081"; //impt! needs a unique value for each record per locid
        myACHTransRecord1.originatorName="MYCOMPANY";
        myACHTransRecord1.transactionCode="PPD"; //PPD,CCD,WEB,TEL,RCK, etc.
        myACHTransRecord1.custTransType="D"; //D for debit, C for credit
        myACHTransRecord1.customerID="CUSTID123-1";
        myACHTransRecord1.customerName="DOE, JOHN"; //Lastname, Firstname - all caps
    }
}
```

```

myACHTransRecord1.customerRoutingNo="987654320"; //9-digit Bank Routing No
myACHTransRecord1.customerAcctNo="00332358882"; //Bank Account No
myACHTransRecord1.customerAcctType="C"; //C for checking, S for savings
myACHTransRecord1.transAmount=100.25;
myACHTransRecord1.checkOrCustID="9166388811";

try
{
    myDTF = DatatypeFactory.newInstance();
    myACHTransRecord1.checkOrTransDate=myDTF.newXMLGregorianCalendar("2010-03-24T00:00:00");
    myACHTransRecord1.effectiveDate=myDTF.newXMLGregorianCalendar("2010-03-24T00:00:00");
}
catch (DatatypeConfigurationException e)
{
    jTextField1.setText("Date exception caught: " + e);
}

myACHTransRecord1.memo="FirstPay";
myACHTransRecord1.opCode="S"; //S for single, R for recurring
myACHTransRecord1.accountSet="1"; //T$$ assigned

//Record 2
myACHTransRecord2.sss="TST"; //T$$ assigned
myACHTransRecord2.locID="9505"; //T$$ assigned
myACHTransRecord2.frontEndTrace="JJ-00082"; //impt! needs a unique value for each record per locid
myACHTransRecord2.originatorName="MYCOMPANY";
myACHTransRecord2.transactionCode="PPD"; //PPD,CCD,WEB,TEL,RCK, etc.
myACHTransRecord2.custTransType="D"; //D for debit, C for credit
myACHTransRecord2.customerID="CUSTID123-2";
myACHTransRecord2.customerName="SMITH, JANET"; //Lastname, Firstname - all caps
myACHTransRecord2.customerRoutingNo="987654320"; //9-digit Bank Routing No
myACHTransRecord2.customerAcctNo="00332358882"; //Bank Account No
myACHTransRecord2.customerAcctType="C"; //C for checking, S for savings
myACHTransRecord2.transAmount=500.25;
myACHTransRecord2.checkOrCustID="9166388811";

try
{
    myDTF = DatatypeFactory.newInstance();
    myACHTransRecord2.checkOrTransDate=myDTF.newXMLGregorianCalendar("2010-03-24T00:00:00");
    myACHTransRecord2.effectiveDate=myDTF.newXMLGregorianCalendar("2010-03-24T00:00:00");
}
catch (DatatypeConfigurationException e)
{
    jTextField1.setText("Date exception caught: " + e);
}

myACHTransRecord2.memo="FirstPay";
myACHTransRecord2.opCode="S"; //S for single, R for recurring
myACHTransRecord2.accountSet="1"; //T$$ assigned

//Record 3
myACHTransRecord3.sss="TST"; //T$$ assigned

```

```

myACHTransRecord3.locID="9505"; //T$$ assigned
myACHTransRecord3.frontEndTrace="JJ-00083"; //impt! needs a unique value for each record per locid
myACHTransRecord3.originatorName="MYCOMPANY";
myACHTransRecord3.transactionCode="PPD"; //PPD,CCD,WEB,TEL,RCK, etc.
myACHTransRecord3.custTransType="C"; //D for debit, C for credit
myACHTransRecord3.customerID="CUSTID123-3";
myACHTransRecord3.customerName="YOUNG, JOE"; //Lastname, Firstname - all caps
myACHTransRecord3.customerRoutingNo="987654320"; //9-digit Bank Routing No
myACHTransRecord3.customerAcctNo="00332358882"; //Bank Account No
myACHTransRecord3.customerAcctType="C"; //C for checking, S for savings
myACHTransRecord3.transAmount=350.25;
myACHTransRecord3.checkOrCustID="9166388811";

```

```

try
{
    myDTF = DatatypeFactory.newInstance();
    myACHTransRecord3.checkOrTransDate=myDTF.newXMLGregorianCalendar("2010-03-24T00:00:00");
    myACHTransRecord3.effectiveDate=myDTF.newXMLGregorianCalendar("2010-03-24T00:00:00");
}
catch (DatatypeConfigurationException e)
{
    jTextField1.setText("Date exception caught: " + e);
}

```

```

myACHTransRecord3.memo="FirstPay";
myACHTransRecord3.opCode="S"; //S for single, R for recurring
myACHTransRecord3.accountSet="1"; //T$$ assigned

```

```

//add ACHTransRecords to ACHFile
myACHFile.achRecords.getACHTransRecord().add(myACHTransRecord1);
myACHFile.achRecords.getACHTransRecord().add(myACHTransRecord2);
myACHFile.achRecords.getACHTransRecord().add(myACHTransRecord3);

```

```

//call SendACHTransBatch method
myACHTransResult = myACHWS.getACHWorksWSSoap().sendACHTransBatch(myCompanyInfo, myACHFile);
jTextField1.setText(myACHTransResult.status + ", " + myACHTransResult.details);

```

```

//list errors if there are any
if (myACHTransResult.totalNumErrors>0)
{
    DefaultListModel myList = new DefaultListModel();

    jList1.setVisibleRowCount(myACHTransResult.totalNumErrors);
    for (int i=0; i<myACHTransResult.totalNumErrors; i++)
        myList.addElement(myACHTransResult.errors.string.get(i));

    jList1.setModel(myList);
}
}
catch (Exception e)
{
    jTextField1.setText("Exception caught: " + e);
}

```



```
}  
}
```

getachreturns

```
private void jButton4MouseClicked(java.awt.event.MouseEvent evt) {  
    //need to 'import' the following:  
    //import java.util.*;  
    //import java.text.*;  
    //import javax.swing.DefaultListModel;  
  
    try  
    {  
        //instances  
        ACHWorksWS myACHWS = new ACHWorksWS();  
        CompanyInfo myCompanyInfo = new CompanyInfo();  
        ACHReturns myACHReturns = new ACHReturns();  
  
        //assign values for myCompanyInfo  
        myCompanyInfo.sss="TST";  
        myCompanyInfo.locID="9505";  
        myCompanyInfo.company="THAT COMPANY";  
        myCompanyInfo.companyKey="RICO";  
  
        //call GetACHReturns method  
        myACHReturns = myACHWS.getACHWorksWS SOAP().getACHReturns(myCompanyInfo);  
        jTextField1.setText(myACHReturns.status + ", " + myACHReturns.details);  
  
        //list ACHReturns if there are any  
        if (myACHReturns.totalNumRecords>0)  
        {  
            DefaultListModel myList = new DefaultListModel();  
  
            jList1.setVisibleRowCount(myACHReturns.totalNumRecords);  
            for (int i=0; i<myACHReturns.totalNumRecords; i++)  
                myList.addElement("FrontEndTrace:" +  
                    myACHReturns.achReturnRecords.getACHReturnRecord().get(i).frontEndTrace +  
                    ", EffectiveDate:" +  
                    myACHReturns.achReturnRecords.getACHReturnRecord().get(i).effectiveDate.toString() +  
                    ", Name:" + myACHReturns.achReturnRecords.getACHReturnRecord().get(i).customerName +  
                    ", Amount:" +  
                    myACHReturns.achReturnRecords.getACHReturnRecord().get(i).transAmount.toString() +  
                    ", ResponseCode:" +  
                    myACHReturns.achReturnRecords.getACHReturnRecord().get(i).responseCode +  
                    ", ActionDetail:" +  
                    myACHReturns.achReturnRecords.getACHReturnRecord().get(i).actionDetail +  
                    ", ActionDate:" +  
                    myACHReturns.achReturnRecords.getACHReturnRecord().get(i).actionDate.toString());  
  
            jList1.setModel(myList);  
        }  
    }  
}
```

```

}
catch (Exception e)
{
    jTextField1.setText("Exception caught: " + e);
}
}

```

getachreturnshist

```

private void jButton5MouseClicked(java.awt.event.MouseEvent evt) {
    //need to 'import' the following:
    //import java.util.*;
    //import java.text.*;
    //import javax.xml.datatype.DatatypeConfigurationException;
    //import javax.xml.datatype.DatatypeFactory;
    //import javax.xml.datatype.XMLGregorianCalendar;
    //import javax.swing.DefaultListModel;

    try
    {
        //instances
        ACHWorksWS myACHWS = new ACHWorksWS();
        CompanyInfo myCompanyInfo = new CompanyInfo();
        ACHReturns myACHReturns = new ACHReturns();
        DatatypeFactory myDTF;

        //assign values for myCompanyInfo
        myCompanyInfo.sss="TST";
        myCompanyInfo.locID="9505";
        myCompanyInfo.company="THAT COMPANY";
        myCompanyInfo.companyKey="RICO";

        try
        {
            //instance
            myDTF = DatatypeFactory.newInstance();

            //call GetACHReturnsHist method
            myACHReturns =
                myACHWS.getACHWorksWS SOAP().getACHReturnsHist(myCompanyInfo,myDTF.newXMLGregorianCalendar("2010-01-01T00:00:00"),myDTF.newXMLGregorianCalendar("2010-03-20T00:00:00"));
        }
        catch (DatatypeConfigurationException e)
        {
            jTextField1.setText("Date exception caught: " + e);
        }

        //assign portion of result to jTextField
        jTextField1.setText(myACHReturns.status + ", " + myACHReturns.details);

        //list ACHReturns if there are any
        if (myACHReturns.totalNumRecords>0)
        {

```

```

DefaultListModel myList = new DefaultListModel();

jList1.setVisibleRowCount(myACHReturns.totalNumRecords);
for (int i=0; i<myACHReturns.totalNumRecords; i++)
    myList.addElement("FrontEndTrace:" +
        myACHReturns.achReturnRecords.getACHReturnRecord().get(i).frontEndTrace +
            ", EffectiveDate:" +
        myACHReturns.achReturnRecords.getACHReturnRecord().get(i).effectiveDate.toString() +
            ", Name:" +
        myACHReturns.achReturnRecords.getACHReturnRecord().get(i).customerName +
            ", Amount:" +
        myACHReturns.achReturnRecords.getACHReturnRecord().get(i).transAmount.toString() +
            ", ResponseCode:" +
        myACHReturns.achReturnRecords.getACHReturnRecord().get(i).responseCode +
            ", ActionDetail:" +
        myACHReturns.achReturnRecords.getACHReturnRecord().get(i).actionDetail +
            ", ActionDate:" +
        myACHReturns.achReturnRecords.getACHReturnRecord().get(i).actionDate.toString());

    jList1.setModel(myList);
}
}
catch (Exception e)
{
    jTextField1.setText("Exception caught: " + e);
}
}

```

getresultfile

```

private void jButton6MouseClicked(java.awt.event.MouseEvent evt) {
    //need to 'import' the following:
    //import java.util.*;
    //import java.text.*;
    //import javax.xml.datatype.DatatypeConfigurationException;
    //import javax.xml.datatype.DatatypeFactory;
    //import javax.xml.datatype.XMLGregorianCalendar;
    //import javax.swing.DefaultListModel;

    try
    {
        //instances
        ACHWorksWS myACHWS = new ACHWorksWS();
        CompanyInfo myCompanyInfo = new CompanyInfo();
        ResultFile myResultFile = new ResultFile();
        DatatypeFactory myDTF;

        //assign values for myCompanyInfo
        myCompanyInfo.sss="TST";
        myCompanyInfo.locID="9505";
        myCompanyInfo.company="THAT COMPANY";
        myCompanyInfo.companyKey="RICO";
    }
}

```

```

try
{
    //instance
    myDTF = DatatypeFactory.newInstance();

    //call GetResultFile method
    myResultFile =
        myACHWS.getACHWorksWS SOAP().getResultFile(myCompanyInfo,myDTF.newXMLGregorianCalendar("2010-
        03-25T00:00:00"),myDTF.newXMLGregorianCalendar("2010-03-26T00:00:00"));
}
catch (DatatypeConfigurationException e)
{
    jTextField1.setText("Date exception caught: " + e);
}

//assign portion of result to jTextField
jTextField1.setText(myResultFile.status + ", " + myResultFile.details);

//list past connection results if there are any
if (myResultFile.totalResultRecords>0)
{
    DefaultListModel myList = new DefaultListModel();

    jList1.setVisibleRowCount(myResultFile.totalResultRecords);
    for (int i=0; i<myResultFile.totalResultRecords; i++)
    {
        myList.addElement("DateTime:" +
            myResultFile.transResults.getTransResult().get(i).callDateTime.toString() +
            ", Method" + myResultFile.transResults.getTransResult().get(i).callMethod +
            ", Status" + myResultFile.transResults.getTransResult().get(i).status +
            ", FileName:" + myResultFile.transResults.getTransResult().get(i).fileName);
        for (int j=0; j<myResultFile.transResults.getTransResult().get(i).totalNumErrors; j++)
            myList.addElement(">" +
                myResultFile.transResults.getTransResult().get(i).errors.getString().get(j));
    }

    jList1.setModel(myList);
}
}
catch (Exception e)
{
    jTextField1.setText("Exception caught: " + e);
}
}

```

geterrorfile

```

private void jButton7MouseClicked(java.awt.event.MouseEvent evt) {
    //need to 'import' the following:
    //import java.util.*;
    //import java.text.*;
    //import javax.xml.datatype.DatatypeConfigurationException;
    //import javax.xml.datatype.DatatypeFactory;

```

```

//import javax.xml.datatype.XMLGregorianCalendar;
//import javax.swing.DefaultListModel;

try
{
    //instances
    ACHWorksWS myACHWS = new ACHWorksWS();
    CompanyInfo myCompanyInfo = new CompanyInfo();
    ErrorFile myErrorFile = new ErrorFile();
    DatatypeFactory myDTF;

    //assign values for myCompanyInfo
    myCompanyInfo.sss="TST";
    myCompanyInfo.locID="9505";
    myCompanyInfo.company="THAT COMPANY";
    myCompanyInfo.companyKey="RICO";

    try
    {
        //instance
        myDTF = DatatypeFactory.newInstance();

        //call GetResultFile method
        myErrorFile =
            myACHWS.getACHWorksWS SOAP().getErrorFile(myCompanyInfo,myDTF.newXMLGregorianCalendar("2010-
            03-25T00:00:00"),myDTF.newXMLGregorianCalendar("2010-03-26T00:00:00"));
    }
    catch (DatatypeConfigurationException e)
    {
        jTextField1.setText("Date exception caught: " + e);
    }

    //assign portion of result to jTextField
    jTextField1.setText(myErrorFile.status + ", " + myErrorFile.details);

    //list past connection results if there are any
    if (myErrorFile.totalErrorRecords>0)
    {
        DefaultListModel myList = new DefaultListModel();

        jList1.setVisibleRowCount(myErrorFile.totalErrorRecords);
        for (int i=0; i<myErrorFile.totalErrorRecords; i++)
        {
            myList.addElement("DateTime:" +
                myErrorFile.errorRecords.getErrorRecord().get(i).callDateTime.toString() +
                ", Method" + myErrorFile.errorRecords.getErrorRecord().get(i).callMethod +
                ", FileName:" + myErrorFile.errorRecords.getErrorRecord().get(i).fileName +
                ", No Of Errors:" +
                myErrorFile.errorRecords.getErrorRecord().get(i).totalNumErrors);
            for (int j=0; j<myErrorFile.getErrorRecords().getErrorRecord().get(i).totalNumErrors; j++)
                myList.addElement(">" +
                    myErrorFile.errorRecords.getErrorRecord().get(i).errors.getString().get(j));
        }
    }
}

```

```
        jList1.setModel(myList);
    }
}
catch (Exception e)
{
    jTextField1.setText("Exception caught: " + e);
}
}
```