



# **DEVELOPER GUIDE**

## **PART C – DEVELOPMENT STEPS**

This is a redacted version of the SIDES Developer Guide. For development, log into the Members Site and obtain the latest version

Version: 6.0

**January 19, 2022**

Copyright © 2022, National Association of State Workforce Agencies.  
All Rights Reserved.

### Revision History

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
8/20/2010	1.0	Version 1.0	SIDES Team
9/27/2010	1.1	<p>Updated: Part C – Development Steps.</p> <p>Added discussion on Combined.xsd to document. See Sections 3.3, 3.3.1.4, and 4.6.1.</p> <p>Enhanced the description of a GUID. See Section 4.2.2.</p> <p>Enhanced discussion on certification test data files to indicate that connectors may need to edit certification test data files. See Section 8.1.1.1.1.</p>	SIDES Team
12/09/2010	1.2	<p>Updated: Part C – Development Steps.</p> <p>Under section 9.1, Common Mistakes, two new sections were added.</p> <p>Section 9.1.7 clarifies the interpretation of numeric fields used to store money values.</p> <p>Section 9.1.8 provides guidance on populating the StateEmployerAccountNumber field so employers and TPAs can use this data to look up employer information in their automated systems.</p>	SIDES Team
4/7/2011	2.0	Added Earnings Verification requirements, modified requirements specific to their	SIDES Team

Date	Version	Description	Author
		exchange, added CCB#9 (C-2.3.3.1-4, 2.3.3.1-5, and 2.3.3.1-6, added model connector.	
5/20/2011	2.1	Updated Part C. Added Employer Model Connector; Jax-WS Model Connector and .Net Model Connector; Corrected SOAP header information relating to the Earnings Verification exchange	SIDES Team
11/18/2011	2.2	Updated Part B.  Added section C-2.10 E-Response and added in the requirement to specify the size of the E-Response custom logo to C-2.10.1.  Identified requirements met through the use of the SIDES Model Connector.	SIDES Team
5/2/2012	3	Updated for Monetary & Potential Charges format and reorganized the guide to split part C into parts C and D.	SIDES Team
9/29/2012	3.1	Removed references to SEW and changed to E-Response Added functionality to bundle requests/responses and added associated business rules	SIDES Team
10/21/2012	3.2	<b>REDACTED</b>	SIDES Team

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
7/18/13	3.3	<p>Removed support of BRPT as a separate tool because of its incorporation into the Model Connector</p> <p>Change to Part – B of Developer Guide: Added date business rules to Earnings Verification and Monetary &amp; Potential Charges exchanges.</p> <p>Deprecated Business Rule 511 due to SIDES Operations Committee's decision to make Response Due Date required</p>	SIDES Team
11/14/13	3.4	Change to Part – B of Developer Guide: ensure the separation information error codes are consistent with production software. Corrected business rules 114 and 123.	SIDES Team
3/14/14	3.5	Change to Part – B of Developer Guide: Added Determinations and Decisions Request and Response XSD rules and business rules error codes.	SIDES Team
9/25/14	3.6	<b>REDACTED</b>	SIDES Team
11/17/14	3.7	<b>REDACTED</b>	SIDES Team
08/26/2016	3.8	Release 2 of EV Standard Format and added missing BRPT code to table	SIDES TEAM

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
		in section 2.1 (Part D of the Developer Guide)	
09/28/2016	3.9	Added Benefit Charges	SIDES Team
02/14/2017	4.0	SIDES Logo Updated	SIDES Team
02/20/2019	5.0	Updated for the new release of MPC	SIDES Team
07/05/2019	5.1	Added a new response business rule to Benefit Changes	SIDES Team
07/10/2019	5.2	Added ADF Error Codes	SIDES Team
9/30/2019	5.3	Added additional ADF Error Codes 1216, 1217, 1218 and changed 'Advance Fact Finding' to 'Additional Fact-Finding.' Note that changes to Error Codes for BC and ADF are highlighted in yellow.	SIDES Team
12/10/2019	5.4	Added missed ADF Response error code 1122. Made language for business rules 1122, 1123, and 1124 consistent business rule documentation.	SIDES Team
12/31/2019	5.5	Added BC Business Rule 1038	SIDES Team
1/27/2020	5.6	Corrected error message 317 to be Business Rule violation – EarningsVerificationWeekBeginDate (Date) must be at least 7 days prior to RequestDate and more recent than RequestDate minus 6 years. It was	SIDES Team

Date	Version	Description	Author
		RequestDate minus 1 year.	
1/19/2022	6.0	Refreshed Parts A, B, C, D of the Developer Guide. Updated for 3/23/2022 SI update. Changed the PRR process from the Production Readiness Review to the Production Self-Certification Readiness Review.	SIDES Team

# TABLE OF CONTENTS

<b>TABLE OF CONTENTS</b> .....	<b>7</b>
<b>1. INTRODUCTION</b> .....	<b>10</b>
<b>2. INITIAL SETUP INSTRUCTIONS</b> .....	<b>12</b>
<b>2.1 URLs</b> .....	<b>12</b>
<b>2.2 Public Key Exchange</b> .....	<b>12</b>
<b>3. COLLECT AND ARRANGE DATA</b> .....	<b>13</b>
<b>3.1 Data Requirements</b> .....	<b>13</b>
<b>3.2 XML</b> .....	<b>15</b>
3.2.1 Sample State Separation Request .....	15
3.2.2 Sample Integrated Partner Separation Response .....	16
3.2.3 Sample State Earnings Verification Request .....	19
3.2.4 Sample Integrated Partner Earnings Verification Response .....	19
3.2.5 Sample State Monetary & Potential Charges Request .....	21
3.2.6 Sample Integrated Partner Monetary & Potential Charges Response .....	22
3.2.7 Sample State Determinations & Decisions Request .....	23
3.2.8 Sample Integrated Partner Determinations & Decisions Response .....	24
3.2.9 Sample State Benefit Charges Request .....	25
3.2.10 Sample Integrated Partner Benefit Charges Response .....	28
3.2.11 Sample State Additional Fact-Finding Request .....	32
3.2.12 Sample Integrated Partner Additional Fact-Finding Response .....	33
<b>3.3 XSD</b> .....	<b>35</b>
3.3.1 Separation Information XSD .....	35
3.3.2 Earnings Verification XSD .....	35
3.3.3 Monetary & Potential Charges XSD .....	36
3.3.4 Determinations & Decisions XSD.....	36
3.3.5 Benefit Charges XSD .....	37
3.3.6 Additional Fact-Finding XSD .....	38
3.3.7 Null/Empty Values .....	38
3.3.8 Dates .....	39
3.3.9 MTOM .....	39
3.3.10 Backfilled Data .....	39
<b>3.4 File Size</b> .....	<b>49</b>
<b>3.5 Business Rules</b> .....	<b>49</b>

<b>4.</b>	<b>BUILD THE CONNECTOR: PREPARING THE MESSAGE.....</b>	<b>51</b>
<b>4.1</b>	<b>Messaging Overview – Post and Pull .....</b>	<b>51</b>
4.1.1	Post .....	51
4.1.2	Pull.....	51
4.1.3	SOAP.....	52
<b>4.2</b>	<b>Messaging Concepts .....</b>	<b>52</b>
4.2.1	Unique ID.....	52
4.2.2	File and Record GUIDs.....	53
4.2.3	SOAP Transaction Number.....	54
4.2.4	Broker Record Transaction Number .....	54
4.2.5	Message Codes .....	54
<b>4.3</b>	<b>SOAP Custom Headers.....</b>	<b>55</b>
4.3.1	State Post .....	57
4.3.2	State Pull.....	61
4.3.3	Integrated Partner Post .....	68
4.3.4	Integrated Partner Pull.....	70
<b>4.4</b>	<b>SOAP Payload .....</b>	<b>78</b>
4.4.1	Separation Information .....	78
4.4.2	Earnings Verification .....	93
4.4.3	Monetary & Potential Charges.....	106
4.4.4	Determinations & Decisions .....	119
4.4.5	Benefit Charges.....	131
4.4.6	Additional Fact-Finding .....	150
<b>4.5</b>	<b>SOAP Action .....</b>	<b>163</b>
<b>4.6</b>	<b>WSDL.....</b>	<b>165</b>
4.6.1	WSDL XSD .....	166
4.6.2	State WSDL.....	167
4.6.3	Integrated Partner WSDL.....	170
<b>5.</b>	<b>BUILD THE CONNECTOR: SECURING THE MESSAGE .....</b>	<b>174</b>
<b>6.</b>	<b>CONNECT WITH THE CENTRAL BROKER: SENDING THE MESSAGE .....</b>	<b>175</b>
<b>6.1</b>	<b>Sending a message .....</b>	<b>175</b>
<b>6.2</b>	<b>Sample SOAP message sent .....</b>	<b>175</b>
<b>6.3</b>	<b>Acknowledgements .....</b>	<b>175</b>
<b>6.4</b>	<b>Non-Broker Returns .....</b>	<b>176</b>



<b>7. CONNECT WITH THE CENTRAL BROKER: TESTING CONNECTOR SOFTWARE ...</b>	<b>177</b>
7.1 Connector Responsibility .....	177
<b>8. CONNECT WITH THE CENTRAL BROKER: SELF-CERTIFICATION READINESS REVIEW</b>	<b>179</b>
8.1 Production Readiness .....	179
8.1.1 Production Readiness Information .....	180
<b>9. LESSONS LEARNED, COMMON MISTAKES, THINGS TO REMEMBER, KEY</b>	<b>184</b>
<b>DEVELOPMENT PITFALLS .....</b>	<b>184</b>
<b>9.1 Lessons Learned.....</b>	<b>184</b>
9.1.1 Using the Black Box jar file in a larger program .....	184
9.1.2 Using the Black Box jar to call directly to the State/Integrated Partner Post/Pull methods.....	184
<b>9.2 Common Mistakes.....</b>	<b>185</b>
9.2.1 Invalid To: and/or From: .....	185
9.2.2 Connector Not a Participant.....	185
9.2.3 Invalid SOAP Action.....	185
9.2.4 Incorrect/Missing Security .....	185
9.2.5 Central Broker Not Having Up-to-Date REDACTED Information .....	185
9.2.6 REDACTED on Server Not Accurate.....	185
9.2.7 Interpretation of Standard Format for Money Fields.....	185
9.2.8 State Employer Account Number.....	186
<b>9.3 Things to Remember .....</b>	<b>186</b>
9.3.1 REDACTED.....	186
9.3.2 Existing Business System Modifications.....	186
9.3.3 Error Handling .....	187
9.3.4 Disaster Recovery .....	187
<b>9.4 Key Development Pitfalls .....</b>	<b>187</b>
9.4.1 REDACTED.....	187
9.4.2 REDACTED.....	187
9.4.3 REDACTED.....	187
<b>10. LIST OF TABLES .....</b>	<b>188</b>

## 1. INTRODUCTION

This document is the third part of the comprehensive Developer Guide package. Part C contains technical guidance to assist you in building the connector to communicate with the Central Broker. Please note that it is very important that you have addressed all of the requirements outlined in Part B – Connector Requirements before you start with this part.

The document is composed of the following:

- **Initial Setup Instructions**
- **A – Collect and Arrange Data**
  - Data Requirements
  - XML (includes samples)
  - XSD
  - File Size
  - Business Rules
- **B - Preparing the Message**
  - Messaging Overview – Post, Pull, and Push
  - Messaging Concepts
  - SOAP Customer Headers
  - SOAP Payload
  - SOAP Actions
  - WSDL
- **C - Securing the Message**
  - **REDACTED**
- **D - Sending the Message**
  - Sending a message
  - Sample SOAP Message
  - Acknowledgements
  - Non-broker Returns
- **E - Testing the Connector Software**
  - Connector Responsibility
  - Tools
    - Model Connectors
    - Business Rule Processor Tool (BRPT)
- **F – Connector Self-Certification Readiness Review**
  - Step 1 - Download Self-Certification Documentation
  - Step 2 - Conduct dry-run Self-Certificatin Readiness Review

- Step 3 - Submit the Production Readiness Spreadsheet
- Step 4 – Conduct Self-Certification Readiness Review
- **Common Mistakes, Things to Remember, Key Development Pitfalls**
- **List of Tables**

Each section contains specific information and examples of how to ensure you are able to successfully execute each process.

For the Connector to interface with the SIDES Central Broker, the Connector must obtain their data from their backend system. This initial step is outside of the scope of the developer guide. Once you obtain the data you need to put it into XML as prescribed by the SIDES XSDs. The development steps documented herein will guide your development of a SIDES compliant connector.

## 2. INITIAL SETUP INSTRUCTIONS

To connect to the SIDES Central Broker, the Connector needs to obtain and configure the appropriate URL to access the SIDES test and SIDES production environment. Public keys must be exchanged between the Connector and the Central Broker. The NASWA SIDES team must set up the Connector within the SIDES Admin Site.

### 2.1 URLs

The URLs used by the Connectors for all messaging with the Central Broker are:

- **REDACTED** [Separation Information production]
- **REDACTED** [Separation Information test]
- **REDACTED** [Earnings Verification production]
- **REDACTED** [Earnings Verification test]
- **REDACTED** [Monetary & Potential Charges production]
- **REDACTED** [Monetary & Potential Charges test]
- **REDACTED** [Determinations & Decisions production]
- **REDACTED** [Determinations & Decisions test]
- **REDACTED** [Benefit Charges production]
- **REDACTED** [Benefit Charges test]
- **REDACTED** [Additional Fact-Finding production]
- **REDACTED** [Additional Fact-Finding test]

### 2.2 Public Key Exchange

**REDACTED**

### 3. COLLECT AND ARRANGE DATA

Prior to collecting and arranging data, the Connector must perform data analysis and mapping between their backend system, the Separation Information Exchange Format, the Earnings Verification Exchange Format, the Monetary and Potential Charges Exchange Format, the Determinations & Decisions Exchange Format, the Benefit Charges Exchange Format, and the Additional Fact-Finding Exchange Format (hereafter known only as Exchange Format). Once the analysis is complete, the Connector must extract data from their backend system and generate an XML file, which will be packaged and delivered to the Central Broker. This section: describes SIDES data requirements; introduces XML and XSDs; defines the date data type used by SIDES; introduces how attachments are transmitted to the Central Broker using Message Transmission Optimization Mechanism (MTOM); explains the backfilled data; specifies the SIDES file size; and articulates the business rules that must be followed to participate in SIDES.

#### 3.1 Data Requirements

The data required for the requests and responses is a set of predetermined data elements, each having its own individual requirements along with its interaction with other elements. The request and response data elements (the standard format) can be found in the SIDES Implementation Guide. This Development Steps guide contains a technical discussion of the information provided for each field in the Separation Information Exchange Format, the Earnings Verification Exchange Format, the Monetary & Potential Charges Exchange Format, the Determinations & Decisions Exchange Format, the Benefit Charges Exchange Format, and the Additional Fact-Finding Format.

**Note:** For the latest version of the Separation Information Exchange Format, the Earnings Verification Exchange Format, the Monetary & Potential Charges Exchange Format, the Determinations & Decisions Exchange Format, the Benefit Charges Exchange Format, or the Additional Fact-Finding Format, please visit the SIDES Website (<https://info.uisides.org>) and obtain the SIDES Implementation Guide, which contains these formats.

There are two tables in the Exchange Formats. The first table outlines the data requirements for the Request sent by a State. The second table outlines the data requirements for the Response sent by an Integrated Partner in response to the Request.

Table 1 describes the information specified for each request and response data element listed in the Exchange Format.

The responsibility of the Connector development team to (a) ensure that, given the information in the Exchange Format, the connecting system can populate all of this data with the development team's current back-end systems or future

planned ones and (b) ensure their client can pass all validations and business rules before they deploy to production.

**Table 1 – Exchange Format**

<b>Column Name</b>	<b>Definition</b>
Seq. Number	The “Sequence Number” is the identifier for the data element. It helps identify which data element is being discussed.
Data Element Name	The “Data Element Name” is the name of the data element.
Data Element Description	The “Data Element Description” is the description of the data element.
Type and/or Format	The “Type and/or Format” is the format of the data element (i.e. character, a numeric, a date or a base64Binary data field).
Field Size	The “Field Size” is the size of the data element. Depending on what type of element it is, it may be a maximum size (2000 character string), or it may be the exact size (10 character date).
Field Required / Optional	<p>“Field Required / Optional” tells whether the field must have an answer for the data element. There are three types of data elements: Required, Optional, and Conditional.</p> <ul style="list-style-type: none"> <li>• Required means that an answer must be filled in.</li> <li>• Optional means that an answer should only be filled in if it makes sense for the State or Integrated Partner to fill in that answer.</li> <li>• Conditional means that an answer may be required or optional depending on the answers to other data elements. (See Business Rules for more information.)</li> </ul>
Business Rules	“Business Rules” are the directives that must be met when filling in this data element. These can include the reasons that will make a conditional element required or restrictions on a date field.
Validation	“Validation” is a check on the data to make sure that the data is in a consistent form and can be readable by all participants.
Comments/Notes	“Comments” are general comments on the data element.

Column Name	Definition
Values	The "Values" column contains all the values that a data element may have for those data elements that are restricted to a set of values.

### 3.2 XML

XML was selected as the medium to implement the SIDES Exchange Format specification for the transfer of the data between States and Integrated Partners. XML allows the data to be defined in a clear and concise manner.

States and Integrated Partners must implement clients that can correctly communicate using the SIDES messaging specification, per the current Requirements Baseline, system design, and as elucidated in this guide.

#### 3.2.1 Sample State Separation Request

**REDACTED**

```
<StateSeparationRequest>

  <!--Header-->
  <StateRequestRecordGUID>11000000000000000000000000000001</StateRequestRecord
  GUID>

  <!--State Info-->
  <RequestingStateAbbreviation>VA</RequestingStateAbbreviation>
  <UIOfficeName>BU1</UIOfficeName>
  <UIOfficePhone>5553247787</UIOfficePhone>
  <UIOfficeFax>5553247787</UIOfficeFax>
  <UIOfficeEmailAddress>customerservice@vec.virginia.gov</UIOfficeEmailAddress>
  <FormNumber>1010101010</FormNumber>
  <AllocationRequiredInd>Y</AllocationRequiredInd>

  <!--Employer Information-->
  <StateEmployerAccountNbr>12000120001200012000</StateEmployerAccountNbr>
  <EmployerName>TV Studio Enterprises</EmployerName>
  <FEIN>123456789</FEIN>
  <TypeofEmployerCode>2</TypeofEmployerCode>
  <!--Claimant Information-->
  <SSN>123456666</SSN>
  <ClaimantLastName>Bunny</ClaimantLastName>
  <OtherLastName>Buns</OtherLastName>
  <ClaimantFirstName>Bugs</ClaimantFirstName>
  <ClaimantMiddleInitial>A</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>

  <!--Claim Data-->
  <ClaimEffectiveDate>2021-01-15</ClaimEffectiveDate>
  <ClaimNumber>1234</ClaimNumber>
```

```

<TypeofClaimCode>2</TypeofClaimCode>
<BenefitYearBeginDate>2021-01-04</BenefitYearBeginDate>
<RequestDate>2021-07-15</RequestDate>
<ResponseDueDate>2021-07-30</ResponseDueDate>

<!--Claim Details-->
<ClaimantSepReasonCode>1</ClaimantSepReasonCode>
<ClaimantSepReasonComments>Temporary Layoff</ClaimantSepReasonComments>

<!--Weeks and Wages-->
<WagesWeeksNeededCode>WW</WagesWeeksNeededCode>
<WagesNeededBeginDate>2020-06-05</WagesNeededBeginDate>
<WagesNeededEndDate>2021-02-05</WagesNeededEndDate>
<!-- Attachment-->
<AttachmentOccurrence>
  <UniqueAttachmentId>01</UniqueAttachmentId>
  <DescriptionofAttachmentCode>4</DescriptionofAttachmentCode>
  <TypeofDocument>Separation Request</TypeofDocument>
  <ActionableAttachment>4</ActionableAttachment>
  <AttachmentSize>1500</AttachmentSize>
  <AttachmentData>MA==</AttachmentData>
</AttachmentOccurrence>
</StateSeparationRequest>
</StateSeparationRequestCollection>

```

### 3.2.2 Sample Integrated Partner Separation Response

**REDACTED**

```

<EmployerTPASeparationResponse>

  <!--Header-->
  <StateRequestRecordGUID>11000000000000000000000000000001</StateRequestRecord
  GUID>
  <BrokerRecordTransactionNumber>105232562</BrokerRecordTransactionNumber>

  <!-- Employer Information -->
  <StateEmployerAccountNbr>12000120001200012000</StateEmployerAccountNbr>
  <CorrectedStateEmployerAccountNbr>12000120001200013000</CorrectedStateEmploye
  rAccountNbr>
  <EmployerName>TV Studio Enterprises</EmployerName>
  <CorrectedEmployerName>Bugs Bunny Studios</CorrectedEmployerName>
  <CorrectedFEIN>123456999</CorrectedFEIN>

  <!-- Claimant Information -->
  <SSN>123456666</SSN>
  <OtherSSN>124405696</OtherSSN>
  <ClaimantLastName>Bunny</ClaimantLastName>
  <ClaimantFirstName>Bugs</ClaimantFirstName>
  <ClaimantMiddleInitial>A</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>
  <ClaimantNameWorkedAsForEmployer>Bugs
  Bunny</ClaimantNameWorkedAsForEmployer>

```



```

<!-- Claim Data -->
<ClaimEffectiveDate>2021-01-15</ClaimEffectiveDate>
<ClaimNumber>1234</ClaimNumber>
<RequestDate>2021-07-15</RequestDate>
<ResponseDueDate>2021-07-30</ResponseDueDate>

<!-- Claim Details -->
<FraudInd>Y</FraudInd>
<FraudReasons>1</FraudReasons>
<ContractEmployee1099Ind>Y</ContractEmployee1099Ind>
<SeasonalEmploymentInd>Y</SeasonalEmploymentInd>
<EmployerSepReasonCode>3</EmployerSepReasonCode>
<ClaimantJobTitle>Engineer</ClaimantJobTitle>
<EmployerReportedClaimantFirstDayofWork>2018-03-
05</EmployerReportedClaimantFirstDayofWork>
<EmployerReportedClaimantLastDayofWork>2021-03-
05</EmployerReportedClaimantLastDayofWork>
<SeparationDateInd>Y</SeparationDateInd>

<!-- Discharge & Suspension -->
<DischargeReasonCode>13</DischargeReasonCode>
<FinalIncidentDate>2021-02-19</FinalIncidentDate>
<FinalIncidentReason>Verbal Altercation</FinalIncidentReason>

<!-- Discharge: Physical Verbal Altercation -->
<AltercationOnCompanyPropertyInd>Y</AltercationOnCompanyPropertyInd>
<AltercationOnCompanyTimeInd>Y</AltercationOnCompanyTimeInd>
<ClaimantStartAltercationInd>Y</ClaimantStartAltercationInd>
<ClaimantAvoidAltercationInd>Y</ClaimantAvoidAltercationInd>
<WereThereWitnessesInd>Y</WereThereWitnessesInd>
<WitnessInformationOccurrence>
  <WitnessName>Balaa</WitnessName>
  <WitnessJobTitle>Supervisor</WitnessJobTitle>
</WitnessInformationOccurrence>
<WitnessInformationOccurrence>
  <WitnessName>Witness Number 2</WitnessName>
  <WitnessJobTitle>Supervisor</WitnessJobTitle>
</WitnessInformationOccurrence>
<WhoDischargedName>Shona</WhoDischargedName>
<WhoDischargedTitle>Manager</WhoDischargedTitle>
<DischargeReasonComments>Discharge Reason
Comments</DischargeReasonComments>

<!-- Prior Incident -->
<PriorIncidentInd>Y</PriorIncidentInd>
<PriorIncidentOccurrence>
  <PriorIncidentDate>2020-10-08</PriorIncidentDate>
  <PriorIncidentReason>Failed to complete tasks according to
instructions</PriorIncidentReason>
  <PriorIncidentWarningInd>Y</PriorIncidentWarningInd>
  <PriorIncidentWarningDate>2020-11-14</PriorIncidentWarningDate>

```

<PriorIncidentWarningDescription>Has been informed about not receiving required tasks completion within deadline in written form through email and verbal both</PriorIncidentWarningDescription>  
</PriorIncidentOccurrence>

<!-- Weeks and Wages -->

<TotalEarnedWagesNeededInd>1</TotalEarnedWagesNeededInd>  
<TotalEarnedWages>170000</TotalEarnedWages>  
<TotalWeeksWorkedNeededInd>1</TotalWeeksWorkedNeededInd>  
<TotalWeeksWorked>14</TotalWeeksWorked>  
<WagesEarnedAfterClaimEffectiveDate>56666</WagesEarnedAfterClaimEffectiveDate>  
<NumberOfHoursWorkedAfterClaimEffectiveDate>1920</NumberOfHoursWorkedAfterClaimEffectiveDate>

<!-- Earnings & Schedule -->

<RegularRateofPay>20.8</RegularRateofPay>  
<RateofPayPerPeriod>H</RateofPayPerPeriod>  
<AverageNumberofHoursWorkedperWeek>40</AverageNumberofHoursWorkedperWeek>  
<CustomaryWeeklyWorkSchedule>From 9am to 6pm</CustomaryWeeklyWorkSchedule>

<!-- Remuneration -->

<RemunerationInd>Y</RemunerationInd>  
<Remuneration>  
    <RemunerationTypeCode>2</RemunerationTypeCode>  
    <EmployerAllocationInd>Y</EmployerAllocationInd>  
    <AllocationBeginDate>2021-03-15</AllocationBeginDate>  
    <AllocationEndDate>2021-09-05</AllocationEndDate>  
    <RemunerationPeriodFrequencyCode>B</RemunerationPeriodFrequencyCode>  
    <RemunerationAmountPerPeriod>350</RemunerationAmountPerPeriod>  
    <NumberHoursRepresented>40</NumberHoursRepresented>  
    <DateRemunerationIssued>2021-03-01</DateRemunerationIssued>  
</Remuneration>

<!-- Pension & 401K -->

<ReceivePension401KInd>4</ReceivePension401KInd>

<!-- Attachment -->

<AdditionalSeparationAckInd>Y</AdditionalSeparationAckInd>  
<AttachmentInd>Y</AttachmentInd>  
<AttachmentOccurrence>  
    <UniqueAttachmentId>01</UniqueAttachmentId>  
    <DescriptionofAttachmentCode>2</DescriptionofAttachmentCode>  
    <TypeofDocument>Response to Separation Notice</TypeofDocument>  
    <AttachmentSize>1000</AttachmentSize>  
    <AttachmentData>MA==</AttachmentData>  
</AttachmentOccurrence>

<!-- Preparer Information -->

<PreparerTypeCode>E</PreparerTypeCode>  
<PreparerTelephoneNumberPlusExt>70312340001234</PreparerTelephoneNumberPlusExt>  
>

```

    <PreparerContactName>Veggies</PreparerContactName>
    <PreparerTitle>Human Resources Manager</PreparerTitle>
    <PreparerEmailAddress>hr@abcdtechnologies.com</PreparerEmailAddress>
</EmployerTPASeparationResponse>
</EmployerTPASeparationResponseCollection>

```

### 3.2.3 Sample State Earnings Verification Request

**REDACTED**

```

    <StateEarningsVerificationRequest>
      <StateEarningsVerificationRequestRecordGUID>AAA53000000000000000000000000003</S
tateEarningsVerificationRequestRecordGUID>
      <RequestingStateAbbreviation>ST</RequestingStateAbbreviation>
      <UIOfficeName>Office Name</UIOfficeName>
      <UIOfficePhone>5555555555</UIOfficePhone>
      <UIOfficeFax>5555555554</UIOfficeFax>
      <UIOfficeEmailAddress>james.madison@state.gov</UIOfficeEmailAddress>
      <StateEmployerAccountNbr>1234567890</StateEmployerAccountNbr>
      <FEIN>123456789</FEIN>
      <EmployerName>ACME</EmployerName>
      <SSN>311111334</SSN>
      <ClaimantLastName>Lastname</ClaimantLastName>
      <ClaimantFirstName>Firstname</ClaimantFirstName>
      <ClaimantMiddleInitial>M</ClaimantMiddleInitial>
      <ClaimantSuffix>JR</ClaimantSuffix>
      <EarningsVerificationWeekBeginDate>2010-08-01</EarningsVerificationWeekBeginDate>
      <EarningsVerificationWeekEndDate>2010-08-07</EarningsVerificationWeekEndDate>
      <EarningsVerificationComments>This is a comment field for this Earnings Verification
Request</EarningsVerificationComments>
      <RequestDate>2010-10-14</RequestDate>
      <EarningsStatusCode>3</EarningsStatusCode>
      <TipsStatusCode>2</TipsStatusCode>
      <CommissionStatusCode>2</CommissionStatusCode>
      <BonusStatusCode>3</BonusStatusCode>
      <VacationStatusCode>2</VacationStatusCode>
      <SickLeaveStatusCode>2</SickLeaveStatusCode>
      <HolidayStatusCode>3</HolidayStatusCode>
      <SeveranceStatusCode>3</SeveranceStatusCode>
      <WagesInLieuStatusCode>4</WagesInLieuStatusCode>
      <EarningsVerificationResponseCommentIndicator>1</EarningsVerificationResponseCom
mentIndicator>
      <ResponseDueDate>2010-10-28</ResponseDueDate>
      <EarningsVerificationSourceCode>9</EarningsVerificationSourceCode>
    </StateEarningsVerificationRequest>
</StateEarningsVerificationRequestCollection>

```

### 3.2.4 Sample Integrated Partner Earnings Verification Response

**REDACTED**

```

    <EmployerTPAEarningsVerificationResponse>
      <!-- Backfilled -->

```

```

    <StateEarningsVerificationRequestRecordGUID>AAA53000000000000000000000000000003</S
tateEarningsVerificationRequestRecordGUID>
    <!-- Backfilled -->
    <BrokerRecordTransactionNumber>5447</BrokerRecordTransactionNumber>
    <!-- Backfilled -->
    <RequestDate>2010-09-01</RequestDate>
    <!-- Backfilled -->
    <ResponseDueDate>2010-09-15</ResponseDueDate>
    <!-- Backfilled -->
    <RequestingStateAbbreviation>ST</RequestingStateAbbreviation>
    <!-- Backfilled -->
    <UIOfficeName>Office Name</UIOfficeName>
    <!-- Backfilled -->
    <StateEmployerAccountNbr>1234567890</StateEmployerAccountNbr>
    <!-- Backfilled -->
    <FEIN>123456789</FEIN>
    <CorrectedFEIN>987654321</CorrectedFEIN>
    <!-- Backfilled -->
    <EmployerName>ACME</EmployerName>
    <CorrectedEmployerName>Fly By Night</CorrectedEmployerName>
    <!-- Backfilled -->
    <SSN>311111334</SSN>
    <!-- Backfilled -->
    <ClaimantLastName>Lastname</ClaimantLastName>
    <!-- Backfilled -->
    <ClaimantFirstName>Firstname</ClaimantFirstName>
    <!-- Backfilled -->
    <ClaimantMiddleInitial>M</ClaimantMiddleInitial>
    <!-- Backfilled -->
    <ClaimantSuffix>JR</ClaimantSuffix>
    <ClaimantNameWorkedAsForEmployer>John Q
Public</ClaimantNameWorkedAsForEmployer>
    <!-- Backfilled -->
    <EarningsVerificationWeekBeginDate>2010-08-01</EarningsVerificationWeekBeginDate>
    <!-- Backfilled -->
    <EarningsVerificationWeekEndDate>2010-08-07</EarningsVerificationWeekEndDate>
    <!-- 1 - Claimant works, 20 - Never Employed Here, 21 - TPA does not represent Employer -
-->
    <ClaimantEmployerWorkRelationshipCode>1</ClaimantEmployerWorkRelationshipCode
>
    <!-- 1 Yes, has earnings, 2 - did not have earnings (100% Sales), 9 - No Work -->
    <EmployerEarningsCode>1</EmployerEarningsCode>
    <FirstDayWorkedinPeriod>2010-08-01</FirstDayWorkedinPeriod>
    <!-- 1 for Yes, 2 for No -->
    <StillWorkingCode>2</StillWorkingCode>
    <LastDayWorked>2010-08-02</LastDayWorked>
    <!-- 1 - Layoff, 2 - Fired, 3 - Vol Quit, 4 - Other -->
    <EmployerSepReasonCode>1</EmployerSepReasonCode>
    <!-- When Request = 1 or (2 with Work/Relationship = 20/21 or Earnings Code = 9) -->
    <EarningsVerificationResponseComment>This employee was let go during the time
period</EarningsVerificationResponseComment>

```

```

<DailyEarningsVerification>
  <EVDay>2010-08-01</EVDay>
  <HoursWorked>15:00</HoursWorked>
  <!-- See Request values for required or not for all below -->
  <AmountEarnedForTheDay>500.00</AmountEarnedForTheDay>
  <EarningsPaidDate>2010-08-02</EarningsPaidDate>
</DailyEarningsVerification>
<DailyEarningsVerification>
  <WeekBeginDate>2010-08-08</WeekBeginDate>
  <WeekEndDate>2010-08-14</WeekEndDate>
  <HoursWorked>15:00</HoursWorked>
  <!-- See Request values for required or not for all below -->
  <AmountEarnedForWeek>500.00</AmountEarnedForWeek>
  <EarningsPaidDate>2010-08-02</EarningsPaidDate>
</DailyEarningsVerification>

<!-- E - Employer, T - TPA -->
<PreparerTypeCode>T</PreparerTypeCode>
<PreparerCompanyName>ABC TPA</PreparerCompanyName>
<PreparerTelephoneNumberPlusExt>5555555556</PreparerTelephoneNumberPlusExt>
<PreparerContactName>Mrs Sue Herman</PreparerContactName>
<PreparerTitle>Claims Administrator</PreparerTitle>
<PreparerFaxNbr>5555555557</PreparerFaxNbr>
<PreparerEmailAddress>sue.herman@abctpa.com</PreparerEmailAddress>
<!-- Backfilled -->
<EarningsVerificationSourceCode>9</EarningsVerificationSourceCode>
</EmployerTPAEarningsVerificationResponse>
</EmployerTPAEarningsVerificationResponseCollection>

```

### 3.2.5 Sample State Monetary & Potential Charges Request

**REDACTED**

```

<StateMonetaryChargeRequest>
  <StateMonetaryChargeRequestRecordGUID>e755d30ed7dd4662bc0452e9050c0001</StateM
onetaryChargeRequestRecordGUID>
  <ClaimNumber>123456789012345678</ClaimNumber>
  <FormNumber>form12345678</FormNumber>
  <RequestDate>2019-04-10</RequestDate>
  <ResponseDueDate>2019-05-05</ResponseDueDate>
  <RequestingStateAbbreviation>CA</RequestingStateAbbreviation>
  <UIOfficeName>CA UIOfficeName</UIOfficeName>
  <UIOfficePhone>2101234567</UIOfficePhone>
  <UIOfficeFax>2101234568</UIOfficeFax>
  <UIOfficeEmailAddress>UIOfficeEmailAddress@asdf.gov</UIOfficeEmailAddress>
  <FEIN>999999999</FEIN>
  <StateEmployerAccountNbr>0123456789</StateEmployerAccountNbr>
  <PredecessorEmployerAccountNbr>0123456789</PredecessorEmployerAccountNbr>
  <EmployerName>Test Employer</EmployerName>
  <EmployerAccountType>1</EmployerAccountType>
  <TypeofEmployerCode>1</TypeofEmployerCode>
  <SSN>123000000</SSN>
  <ClaimantLastName>Doe</ClaimantLastName>

```

```

<OtherLastName>Mouse</OtherLastName>
<ClaimantFirstName>Jane</ClaimantFirstName>
<ClaimantMiddleInitial>A</ClaimantMiddleInitial>
<ClaimantSuffix>Jr</ClaimantSuffix>
<Dependents>2</Dependents>
<ClaimantSepReasonCode>2</ClaimantSepReasonCode>
<BenefitYearBeginDate>2013-07-23</BenefitYearBeginDate>
<BasePeriodStartDate>2010-07-24</BasePeriodStartDate>
<BasePeriodEndDate>2011-07-23</BasePeriodEndDate>
<NumberEmployersInBasePeriod>1</NumberEmployersInBasePeriod>
<BasePeriodWagesAllEmployers>42000.00</BasePeriodWagesAllEmployers>
<BasePeriodWagesThisEmployer>36000.00</BasePeriodWagesThisEmployer>
<WeeklyBenefitAmount>1000.00</WeeklyBenefitAmount>
<MaximumBenefitAmount>60000.00</MaximumBenefitAmount>
<TotalWeeksOfBenefits>52</TotalWeeksOfBenefits>
<PercentChargableEmployer>100.00</PercentChargableEmployer>
<TotalEmployerCharge>50000.00</TotalEmployerCharge>
<EmployerChargeOccurrence>
  <BeginWeekNumber>1</BeginWeekNumber>
  <EndWeekNumber>1</EndWeekNumber>
  <TotalWeeklyEmployerCharge>1000.00</TotalWeeklyEmployerCharge>
  <PotentialWeeksEmployerCharge>10000.00</PotentialWeeksEmployerCharge>
</EmployerChargeOccurrence>
<EmployerChargeOccurrence>
  <BeginWeekNumber>2</BeginWeekNumber>
  <EndWeekNumber>2</EndWeekNumber>
  <TotalWeeklyEmployerCharge>1200.00</TotalWeeklyEmployerCharge>
  <PotentialWeeksEmployerCharge>12000.00</PotentialWeeksEmployerCharge>
</EmployerChargeOccurrence>
<EmployerChargeOccurrence>
  <BeginWeekNumber>3</BeginWeekNumber>
  <EndWeekNumber>3</EndWeekNumber>
  <TotalWeeklyEmployerCharge>1000.00</TotalWeeklyEmployerCharge>
  <PotentialWeeksEmployerCharge>10000.00</PotentialWeeksEmployerCharge>
</EmployerChargeOccurrence>
<WagesInFirstQuarter>9000.00</WagesInFirstQuarter>
<WagesInSecondQuarter>9000.00</WagesInSecondQuarter>
<WagesInThirdQuarter>9000.00</WagesInThirdQuarter>
<WagesInFourthQuarter>9000.00</WagesInFourthQuarter>
<InitialClaimDate>2018-11-12</InitialClaimDate>
<RequestforReliefofChargesAllowedInd>Y</RequestforReliefofChargesAllowedInd>
</StateMonetaryChargeRequest>
</StateMonetaryChargeRequestCollection>

```

### 3.2.6 Sample Integrated Partner Monetary & Potential Charges Response

**REDACTED**

```

<EmployerTPAMonetaryChargeResponse>
  <StateMonetaryChargeRequestRecordGUID>e755d30ed7dd4662bc0452e9050c0001</StateM
onetaryChargeRequestRecordGUID>

```

```

<BrokerRecordTransactionNumber>641080</BrokerRecordTransactionNumber>
<ClaimNumber>123456789012345678</ClaimNumber>
<RequestDate>2019-04-10</RequestDate>
<ResponseDueDate>2019-05-05</ResponseDueDate>
<CorrectedFEIN>999999999</CorrectedFEIN>
<StateEmployerAccountNbr>0123456789</StateEmployerAccountNbr>
<CorrectedStateEmployerAccountNbr>2123456789</CorrectedStateEmployerAccountNbr>
<EmployerName>Test Employer</EmployerName>
<CorrectedEmployerName>Test Employer 2</CorrectedEmployerName>
<SSN>123000000</SSN>
<OtherSSN>223001234</OtherSSN>
<ClaimantLastName>Doe</ClaimantLastName>
<ClaimantFirstName>Jane</ClaimantFirstName>
<ClaimantMiddleInitial>A</ClaimantMiddleInitial>
<ClaimantSuffix>Jr</ClaimantSuffix>
<ClaimantNameWorkedAsForEmployer>Donuld
Duck</ClaimantNameWorkedAsForEmployer>
<SeasonalEmploymentInd>Y</SeasonalEmploymentInd>
<EmployerReportedDependents>2</EmployerReportedDependents>
<RequestReliefOfChargesInd>N</RequestReliefOfChargesInd>
<EmployerCorrectedWageforFirstQuarter>1000.0</EmployerCorrectedWageforFirstQuarter>
<EmployerCorrectedWageforSecondQuarter>1000.0</EmployerCorrectedWageforSecondQu
arter>
<EmployerCorrectedWageforThirdQuarter>1000.0</EmployerCorrectedWageforThirdQuarter>
<EmployerCorrectedWageforFourthQuarter>1000.0</EmployerCorrectedWageforFourthQuart
er>
<AttachmentOccurrence>
  <DescriptionOfAttachmentCode>1</DescriptionOfAttachmentCode>
  <TypeOfDocument>Test Response Attachment</TypeOfDocument>
  <UniqueAttachmentId>01</UniqueAttachmentId>
  <AttachmentSize>1</AttachmentSize>
  <AttachmentData>MA==</AttachmentData>
</AttachmentOccurrence>
<PreparerTypeCode>E</PreparerTypeCode>
<PreparerCompanyName>PreparerCompanyName</PreparerCompanyName>
<PreparerTelephoneNumberPlusExt>2101234567</PreparerTelephoneNumberPlusExt>
<PreparerContactName>Joe Preparer</PreparerContactName>
<PreparerTitle>Preparer Tester</PreparerTitle>
<PreparerFaxNbr>2101234567</PreparerFaxNbr>
<PreparerEmailAddress>prep@asdf.asdf</PreparerEmailAddress>
</EmployerTPAMonetaryChargeResponse>
</EmployerTPAMonetaryChargeResponseCollection>

```

### 3.2.7 Sample State Determinations & Decisions Request

#### REDACTED

```

<StateDeterminationDecisionRequest>
  <StateDeterminationDecisionRequestRecordGUID>101000000000000000000000000000001111</S
tateDeterminationDecisionRequestRecordGUID>

```

```

<RequestDate>2014-02-27</RequestDate>
<RequestingStateAbbreviation>ST</RequestingStateAbbreviation>
<SSN>101011111</SSN>
<ClaimantLastName>Doe</ClaimantLastName>
<OtherLastName>Simpson</OtherLastName>
<ClaimantFirstName>Homer</ClaimantFirstName>
<ClaimEffectiveDate>2013-01-21</ClaimEffectiveDate>
<ClaimNumber>99991111999919</ClaimNumber>
<BenefitYearBeginDate>2012-11-21</BenefitYearBeginDate>
<StateEmployerAccountNbr>123456789</StateEmployerAccountNbr>
<FEIN>123000656</FEIN>
<EmployerName>JC Penney Inc</EmployerName>
<UIOfficeName>CA UI Claims</UIOfficeName>
<DeterminationOrDecisionInd>10</DeterminationOrDecisionInd>
<DeterminationOrDecisionDate>2013-03-21</DeterminationOrDecisionDate>
<DeterminationOrDecisionResult>10</DeterminationOrDecisionResult>
<EmployerChargableForBenefitsPaid>Y</EmployerChargableForBenefitsPaid>
<AgentResponseIndicator>1</AgentResponseIndicator>
<EmpWorksiteIndicator>1</EmpWorksiteIndicator>
<NotAvailableIndicator>1</NotAvailableIndicator>
<InterpreterIndicator>1</InterpreterIndicator>
<SpecialNeedsIndicator>1</SpecialNeedsIndicator>
<AttachmentOccurrence>
  <DescriptionofAttachmentCode>1</DescriptionofAttachmentCode>
  <TypeofDocument>Test Request Attachment</TypeofDocument>
  <UniqueAttachmentId>01</UniqueAttachmentId>
  <ActionableAttachment>11</ActionableAttachment>
  <AttachmentSize>1</AttachmentSize>
  <AttachmentData>e1xydGYxXGfKZWZsYW5nMTAyN=</AttachmentData>
</AttachmentOccurrence>
<AppealableDeterminationOrDecision>1</AppealableDeterminationOrDecision>
<LastDateToAppeal>2014-02-27</LastDateToAppeal>
</StateDeterminationDecisionRequest>
</StateDeterminationDecisionRequestCollection>

```

### 3.2.8 Sample Integrated Partner Determinations & Decisions Response

**REDACTED**

```

<EmployerTPADeterminationDecisionResponse>
  <StateDeterminationDecisionRequestRecordGUID>121300000000000000000000000011111</S
tateDeterminationDecisionRequestRecordGUID>
  <BrokerRecordTransactionNumber>8510</BrokerRecordTransactionNumber>
  <RequestDate>2014-06-15</RequestDate>
  <LastDateToAppeal>2014-07-02</LastDateToAppeal>
  <RequestingStateAbbreviation>ST</RequestingStateAbbreviation>
  <SSN>011311111</SSN>
  <ClaimantLastName>Doe</ClaimantLastName>
  <OtherLastName>Simpson</OtherLastName>
  <ClaimantFirstName>Homer</ClaimantFirstName>
  <ClaimEffectiveDate>2013-07-23</ClaimEffectiveDate>
  <ClaimNumber>99991111999919</ClaimNumber>
  <BenefitYearBeginDate>2013-05-23</BenefitYearBeginDate>

```



```

<StateEmployerAccountNbr>123456789</StateEmployerAccountNbr>
<FEIN>123000656</FEIN>
<EmployerName>JC Penney Inc</EmployerName>
<DeterminationOrDecisionInd>12</DeterminationOrDecisionInd>
<DeterminationOrDecisionDate>2013-09-20</DeterminationOrDecisionDate>

<EmployerRequestingRedetOrAppealOfDeterminationOrDecision>Y</EmployerRequesting
RedetOrAppealOfDeterminationOrDecision>
<AppealReasonIndicator>10</AppealReasonIndicator>
<EmployerReasonRequestingRedetOrAppeal>Test the
comments</EmployerReasonRequestingRedetOrAppeal>
<AttachmentOccurrence>
  <DescriptionofAttachmentCode>1</DescriptionofAttachmentCode>
  <TypeofDocument>Test Request Attachment</TypeofDocument>
  <UniqueAttachmentId>01</UniqueAttachmentId>
  <AttachmentSize>1</AttachmentSize>
  <AttachmentData>e lxydGYxXGfKZWZsYW5nMTAyNVxhbnfX0=</AttachmentData>
</AttachmentOccurrence>
<AgentType>2</AgentType>
<AgentAttorneyName>Roger Smith</AgentAttorneyName>
<AgentAttorneyAddress1>1 Penn State Ave</AgentAttorneyAddress1>
<AgentAttorneyAddress2>#100</AgentAttorneyAddress2>
<AgentAttorneyCity>Philadelphia</AgentAttorneyCity>
<AgentAttorneyState>PA</AgentAttorneyState>
<AgentAttorneyZip>06877</AgentAttorneyZip>
<AgentAttorneyTelephoneNumberPlusExt>2132122122</AgentAttorneyTelephoneNumberP
lusExt>
<InterpreterLanguage>Interpret in Spanish</InterpreterLanguage>
<SpecialNeeds>No Special Needs Required</SpecialNeeds>
<PreparerTypeCode>E</PreparerTypeCode>
<PreparerTelephoneNumberPlusExt>2132122122</PreparerTelephoneNumberPlusExt>
<PreparerContactName>Dean Smity</PreparerContactName>
<PreparerTitle>Manager</PreparerTitle>
<PreparerEmailAddress>snooty@snoots.com</PreparerEmailAddress>
</EmployerTPADeterminationDecisionResponse>
</EmployerTPADeterminationDecisionResponseCollection>

```

### 3.2.9 Sample State Benefit Charges Request

#### REDACTED

```

<StateBenefitChargesRequest>
  <!--Required: 32 characters-->
  <StateChargeRecordGUID>00000000000000000000000000000802
</StateChargeRecordGUID>
  <!--Required: yyyy-mm-dd: Must be previous date, or less than current date + 6
months-->
  <NoticeDate>2016-11-08</NoticeDate>
  <!--Required: Y = Yes, N = No-->
  <DueDateRequiredInd>Y</DueDateRequiredInd>
  <!--Conditional: yyyy-mm-dd: Required if DueDateRequiredInd is Y, -->
  <LastDayToRespond>2016-11-29</LastDayToRespond><!--Required: 2 Characters-->
  <ChargingStateAbbreviation>ST</ChargingStateAbbreviation><!--Required: Up to 25
characters-->

```

<UIOfficeName>Test Office 2</UIOfficeName><!--Optional: up to 60 characters-->  
 <UIOfficeEmailAddress>test@email.com</UIOfficeEmailAddress><!--Optional: Up to  
 15 digits-->  
 <UIOfficePhone>3014310826</UIOfficePhone><!--Optional: Up to 10 digits-->  
 <UIOfficeFax>3014310827</UIOfficeFax><!--Required: 1 Digit: 1 = Reserve Ratio, 2 =  
 Benefit Ratio (may also have a reserve ratio component)--><!--3 = Benefit Wage Ratio, 4 =  
 Payroll Variation-->  
 <ExperienceRatingMethod>2</ExperienceRatingMethod>  
 <!--Required: Up to 20 characters-->  
 <StateEmployerAccountNbr>8000001</StateEmployerAccountNbr>  
 <!--Required: Up to 100 characters-->  
 <EmployerName>Second Test Company</EmployerName>  
 <!--Required: 9 Digits-->  
 <FEIN>301416817</FEIN>  
 <!--Required: yyyy-mm-dd: Must be less than or equal to NoticeDate minus seven  
 days and equal to or greater than NoticeDate minus 13 months, 7 days-->  
 <ChargeStatementStartDate>2016-11-01</ChargeStatementStartDate>  
 <!--Required: yyyy-mm-dd: Greater or equal to 7, and less than 93 days after  
 StartDate-->  
 <ChargeStatementEndDate>2016-11-29</ChargeStatementEndDate>  
 <!--Required: 1 = taxable employer, 2 = reimbursable employer-->  
 <EmployerAccountType>1</EmployerAccountType>  
 <!--Required: Up to 7 numbers: 1 or larger-->  
 <NumberOfIndividualSSNsWithCharges>2</NumberOfIndividualSSNsWithCharges>  
 <!--Required: 15 Number, 2 decimals automatic-->  
 <TotalDollarAmountChargedToEmployer>5400.00  
 </TotalDollarAmountChargedToEmployer>  
  
 <!--Part 2: Required: Repeatable 1-n times, n = NumberOfIndividualSSNsWithCharges-  
 -->  
 <SSNBYBChargeProgramCode><!--Required: 9 numbers-->  
 <SSN>216262448</SSN>  
 <!--Required: yyyy-mm-dd: At least 7 days prior to NoticeDate, not more than 6  
 years before NoticeDate-->  
 <BenefitYearBeginDate>2016-09-19</BenefitYearBeginDate><!--Conditional:  
 Required for each SSNBYBChargeProgramCode: 1 Number: 1 = Regular State UI Benefits--><!--2 =  
 Combined Wage Claim, 3 = Federal Extended Benefits, 4 = State Extended Benefits, 5 =  
 Combined Wage Claim extended benefits-->  
 <ProgramCode>1</ProgramCode>  
 <!--Optional: Up to 18 characters-->  
 <ClaimNumber>802</ClaimNumber>  
 <!--Required: Up to 30 characters-->  
 <ClaimantLastName>Adams</ClaimantLastName>  
 <!--Optional: Up to 30 characters-->  
 <OtherLastName>Smith</OtherLastName>  
 <!--Required: Up to 25 characters-->  
 <ClaimantFirstName>John</ClaimantFirstName>  
 <!--Required: 1 character-->  
 <ClaimantMiddleInitial>A</ClaimantMiddleInitial>  
 <!--Optional: 1 number: 1 = Last, Base period, and Lag quarter employer, 2 =  
 Last employer, 3 = Base periodd employer--><!--4 = Lag quarter employer, 5 = Last and Base

Period employer, 6 = Last and Lag Quarter employer, 7 = Base period and Lag quarter employer-->

```
<TypeofEmployerCode>5</TypeofEmployerCode>
<!--Required: yyyy-mm-dd: Must be equal to or more recent than
BenefitYearBeginDate minus 79 weeks, and at least 52 weeks prior to BenefitYearBeginDate-->
<BasePeriodStartDate>2015-07-07</BasePeriodStartDate>
<!--Required: yyyy-mm-dd: Must equal BasePeriodStartDate plus 53 weeks-->
<BasePeriodEndDate>2016-07-12</BasePeriodEndDate>
<!--Optional: Up to 15 numbers, 2 decimals automatic-->
<BasePeriodWages>42807.69</BasePeriodWages>
<!--Required: Up to 15 numbers, 2 decimals: Negative numbers are for credits-->
```

```
>
<TotalAmountChargedToEmployer>2700</TotalAmountChargedToEmployer>
<!--Required: Up to 2 numbers: 1 - 33-->
<TotalNumberOfWeeksChargedToEmployerAccount>9
</TotalNumberOfWeeksChargedToEmployerAccount>
<!--Optional: Up to 2000 characters-->
<BenefitChargeComment>Charged at 300.00 per
week</BenefitChargeComment>
</SSNBYBChargeProgramCode>
```

```
<!--Part 2: Required: Repeatable 1-n times, n = NumberOfIndividualSSNsWithCharges-->
<SSNBYBChargeProgramCode><!--Required: 9 numbers-->
<SSN>498653795</SSN>
<!--Required: yyyy-mm-dd: At least 7 days prior to NoticeDate, not more than 6
years before NoticeDate-->
<BenefitYearBeginDate>2016-09-19</BenefitYearBeginDate><!--Conditional:
Required for each SSNBYBChargeProgramCode: 1 Number: 1 = Regular State UI Benefits--><!--2 =
Combined Wage Claim, 3 = Federal Extended Benefits, 4 = State Extended Benefits, 5 =
Combined Wage Claim extended benefits-->
```

```
<ProgramCode>2</ProgramCode>
<!--Optional: Up to 18 characters-->
<ClaimNumber>1802</ClaimNumber>
<!--Required: Up to 30 characters-->
<ClaimantLastName>Grey</ClaimantLastName>
<!--Required: Up to 25 characters-->
<ClaimantFirstName>Patrick</ClaimantFirstName>
<!--Required: 1 character-->
<ClaimantMiddleInitial>J</ClaimantMiddleInitial>
<!--Optional: Up to 8 characters-->
<ClaimantSuffix>Jr</ClaimantSuffix>
<!--Optional: 1 number: 1 = Last, Base period, and Lag quarter employer, 2 =
Last employer, 3 = Base period employer--><!--4 = Lag quarter employer, 5 = Last and Base
Period employer, 6 = Last and Lag Quarter employer, 7 = Base period and Lag quarter
employer-->
```

```
<TypeofEmployerCode>1</TypeofEmployerCode>
<!--Required: yyyy-mm-dd: Must be equal to or more recent than
BenefitYearBeginDate minus 79 weeks, and at least 52 weeks prior to BenefitYearBeginDate-->
<BasePeriodStartDate>2015-07-07</BasePeriodStartDate>
<!--yyyy-mm-dd: Must equal BasePeriodStartDate plus 53 weeks-->
<BasePeriodEndDate>2016-07-12</BasePeriodEndDate>
```

```

> <!--Required: Up to 15 numbers, 2 decimals: Negative numbers are for credits-->
<TotalAmountChargedToEmployer>2700</TotalAmountChargedToEmployer>
<!--Required: Up to 2 numbers: 1 - 33-->

<TotalNumberOfWeeksChargedToEmployerAccount>6</TotalNumberOfWeeksChargedTo
EmployerAccount>
<!--Optional: Up to 2000 characters-->
<BenefitChargeComment>450.00 per week</BenefitChargeComment>
</SSNBYBChargeProgramCode>
<!--Required: 1 number: 1 = Yes, 2 = No-->
<AgentResponseIndicator>2</AgentResponseIndicator>
<!--Required: 1 number: 1 = Yes, 2 = No-->
<NotAvailableIndicator>2</NotAvailableIndicator>
<!--Required: 1 number: 1 = Yes, 2 = No-->
<InterpreterIndicator>2</InterpreterIndicator>
<!--Required: 1 number: 1 = Yes, 2 = No-->
<SpecialNeedsIndicator>2</SpecialNeedsIndicator>
<!--Required: 1 number: 1 = Protest, 2 = Appeal, 3 = Question, 4 = Protest/Question, 5 =
Appeal/Question-->
<StateTerminologyUsed>1</StateTerminologyUsed>
</StateBenefitChargesRequest>
</StateBenefitChargesRequestCollection>

```

### 3.2.10 Sample Integrated Partner Benefit Charges Response

#### REDACTED

```

<!--Backfilled-->
><EmployerTPABenefitChargesResponse><StateChargeRecordGUID>0000000000000000000000
000000803</StateChargeRecordGUID><!--Backfilled-->
><BrokerRecordTransactionNumber>6000</BrokerRecordTransactionNumber>
<!--Backfilled--><NoticeDate>2016-11-08</NoticeDate>
<!--Backfilled--><LastDayToRespond>2016-11-29</LastDayToRespond>
<!--Backfilled--><ChargingStateAbbreviation>ST</ChargingStateAbbreviation>
<!--Backfilled--><StateEmployerAccountNbr>8000001</StateEmployerAccountNbr>
<!--Backfilled--><EmployerName>Third Test Company</EmployerName>
<!--Backfilled--><FEIN>301416817</FEIN>
<!--Backfilled--><ChargeStatementStartDate>2016-11-01</ChargeStatementStartDate>
<!--Backfilled--><ChargeStatementEndDate>2016-11-09</ChargeStatementEndDate>
<!--Required: 1-n times--><SSNBYBChargeProgramCode>
<!--Backfilled-->
<SSN>123456789</SSN>
<!--Backfilled-->
<ClaimantLastName>Edwards</ClaimantLastName>
<!--Backfilled-->
<ClaimantFirstName>Jenna</ClaimantFirstName>
<!--Backfilled-->
<ClaimantMiddleInitial>N</ClaimantMiddleInitial>
<!--Backfilled-->
<BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>

```

<!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this company during this period-->  
 <!--11 = Claimant was working for some other employer during this period, 20 = Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not available for work, 40 = Claimant worked for employer less than the time period or earnings required to be a chargeable employer--><!--41 = Charges are for a period of claimant ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was not received, 43 = Determination of eligibility not received-->  
 <!--44 = Employer is exempt from charges due to circumstances of separation, 46 = Credit has been previously identified and approved, but not received-->  
 <!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone ever having worked for them under this name or SSN, 60 = Charges are being questioned due to a pending protest/appeal of eligibility determination, 70 = Employer questions charges for statutory reasons NOT listed above-->  
 <ProgramCode>1</ProgramCode>  
 <ProtestAppealQuestionReasonInd>10</ProtestAppealQuestionReasonInd>  
 <!--Required: Up to 2000 characters-->  
 <EmployerReasonForProtestAppealQuestion>claimant worked here</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode><!--Required: 1-n times-->  
 <SSNBYBChargeProgramCode>  
 <!--Backfilled-->  
 <SSN>456932654</SSN>  
 <!--Backfilled-->  
 <ClaimantLastName>Johns</ClaimantLastName>  
 <!--Backfilled-->  
 <ClaimantFirstName>Jenna</ClaimantFirstName>  
 <!--Backfilled-->  
 <ClaimantMiddleInitial>K</ClaimantMiddleInitial>  
 <!--Backfilled-->  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <!--Backfilled-->  
 <ProgramCode>2</ProgramCode>  
 <!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this company during this period-->  
 <!--11 = Claimant was working for some other employer during this period, 20 = Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not available for work, 40 = Claimant worked for employer less than the time period or earnings required to be a chargeable employer--><!--41 = Charges are for a period of claimant ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was not received, 43 = Determination of eligibility not received-->  
 <!--44 = Employer is exempt from charges due to circumstances of separation, 46 = Credit has been previously identified and approved, but not received-->  
 <!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone ever having worked for them under this name or SSN, 60 = Charges are being questioned due to a pending protest/appeal of eligibility determination, 70 = Employer questions charges for statutory reasons NOT listed above-->  
 <ProtestAppealQuestionReasonInd>51</ProtestAppealQuestionReasonInd>  
 <!--Required: Up to 2000 characters-->  
 <EmployerReasonForProtestAppealQuestion>Jenna Johns has not worked here</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode>

<!--Required: 1-n times-->  
 <SSNBYBChargeProgramCode>  
 <!--Backfilled-->  
 <SSN>203968375</SSN>  
 <!--Backfilled-->  
 <ClaimantLastName>Douglass</ClaimantLastName>  
 <!--Backfilled-->  
 <ClaimantFirstName>Frederick</ClaimantFirstName>  
 <!--Backfilled-->  
 <ClaimantMiddleInitial>A</ClaimantMiddleInitial>  
 <!--Backfilled-->  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <!--Backfilled-->  
 <ProgramCode>3</ProgramCode>  
 <!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this  
 company during this period-->  
 <!--11 = Claimant was working for some other employer during this period, 20 =  
 Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not  
 available for work, 40 = Claimant worked for employer less than the time period or earnings  
 required to be a chargeable employer--><!--41 = Charges are for a period of claimant  
 ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was  
 not received, 43 = Determination of eligibility not received-->  
 <!--44 = Employer is exempt from charges due to circumstances of separation, 46 =  
 Credit has been previously identified and approved, but not received-->  
 <!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone ever  
 having worked for them under this name or SSN, 60 = Charges are being questioned due to a  
 pending protest/appeal of eligibility determination, 70 = Employer questions charges for  
 statutory reasons NOT listed above-->  
 <ProtestAppealQuestionReasonInd>10</ProtestAppealQuestionReasonInd>  
 <!--Required: Up to 2000 characters-->  
 <EmployerReasonForProtestAppealQuestion>Frederick Douglass worked  
 here</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode>

<!--Required: 1-n times-->  
 <SSNBYBChargeProgramCode>  
 <!--Backfilled-->  
 <SSN>217262897</SSN>  
 <!--Backfilled-->  
 <ClaimantLastName>Kerry</ClaimantLastName>  
 <!--Backfilled-->  
 <ClaimantFirstName>Marcia</ClaimantFirstName>  
 <!--Backfilled-->  
 <ClaimantMiddleInitial>B</ClaimantMiddleInitial>  
 <!--Backfilled-->  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <!--Backfilled-->  
 <ProgramCode>4</ProgramCode>  
 <!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this  
 company during this period-->

<!--11 = Claimant was working for some other employer during this period, 20 = Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not available for work, 40 = Claimant worked for employer less than the time period or earnings required to be a chargeable employer--><!--41 = Charges are for a period of claimant ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was not received, 43 = Determination of eligibility not received-->

<!--44 = Employer is exempt from charges due to circumstances of separation, 46 = Credit has been previously identified and approved, but not received-->

<!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone ever having worked for them under this name or SSN, 60 = Charges are being questioned due to a pending protest/appeal of eligibility determination, 70 = Employer questions charges for statutory reasons NOT listed above-->

<ProtestAppealQuestionReasonInd>41</ProtestAppealQuestionReasonInd>

<!--Required: Up to 2000 characters-->

<EmployerReasonForProtestAppealQuestion>See previous  
DD</EmployerReasonForProtestAppealQuestion>  
</SSNBYBChargeProgramCode>

<!--Required: 1-n times-->

<SSNBYBChargeProgramCode>

<!--Backfilled-->

<SSN>217259876</SSN>

<!--Backfilled-->

<ClaimantLastName>Jones</ClaimantLastName>

<!--Backfilled-->

<ClaimantFirstName>Michael</ClaimantFirstName>

<!--Backfilled-->

<ClaimantMiddleInitial>E</ClaimantMiddleInitial>

<!--Backfilled-->

<BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>

<ProgramCode>5</ProgramCode>

<!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this company during this period-->

<!--11 = Claimant was working for some other employer during this period, 20 = Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not available for work, 40 = Claimant worked for employer less than the time period or earnings required to be a chargeable employer--><!--41 = Charges are for a period of claimant ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was not received, 43 = Determination of eligibility not received-->

<!--44 = Employer is exempt from charges due to circumstances of separation, 46 = Credit has been previously identified and approved, but not received-->

<!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone ever having worked for them under this name or SSN, 60 = Charges are being questioned due to a pending protest/appeal of eligibility determination, 70 = Employer questions charges for statutory reasons NOT listed above-->

<ProtestAppealQuestionReasonInd>10</ProtestAppealQuestionReasonInd>

<!--Required: Up to 2000 characters-->

<EmployerReasonForProtestAppealQuestion>claimant worked  
here</EmployerReasonForProtestAppealQuestion>  
</SSNBYBChargeProgramCode>

<!--Conditional: 1 digit: 1 = No, 2 = Agent, 3 = Attorney-->

```

    <AgentType>1</AgentType>
    <!--Required: 1 character: E = Employer, T = Third Party Administrator-->
    <PreparerTypeCode>E</PreparerTypeCode><!--Required: Up to 15 digits-->
    <PreparerTelephoneNumberPlusExt>7036634563</PreparerTelephoneNumberPlusExt>
    <!--Required: Up to 60 characters--><PreparerContactName>Jason
Smith</PreparerContactName>
    <!--Required: Up to 30 characters--><PreparerTitle>HR Manager</PreparerTitle>
    <!--Required: Up to 60 characters-->
><PreparerEmailAddress>jsmith@testco3.com</PreparerEmailAddress>
    </EmployerTPABenefitChargesResponse>

</EmployerTPABenefitChargesResponseCollection>

```

### 3.2.11 Sample State Additional Fact-Finding Request

#### REDACTED

```

<StateAdditionalFactFindingRequest>
  <StateRequestRecordGUID>110000000000000000000000000001</StateRequestRecordGUID>
  <SSN>000110001</SSN>
  <ClaimEffectiveDate>2019-09-10</ClaimEffectiveDate>
  <ClaimNumber>11231</ClaimNumber>
  <StateEmployerAccountNbr>179424876</StateEmployerAccountNbr>
  <EmployerName>Pennies Inc</EmployerName>
  <FEIN>100000001</FEIN>
  <TypeofClaimCode>1</TypeofClaimCode>
  <BenefitYearBeginDate>2016-09-28</BenefitYearBeginDate>
  <RequestingStateAbbreviation>CA</RequestingStateAbbreviation>
  <UIOfficeName>BU1</UIOfficeName>
  <UIOfficePhone>5553247787</UIOfficePhone>
  <UIOfficeFax>5553247787</UIOfficeFax>
  <AdjudicatorIdentifier>ADJ101</AdjudicatorIdentifier>
  <AdjudicatorName>Linda Hoffman</AdjudicatorName>
  <AdjudicatorPhone>5553247787</AdjudicatorPhone>
  <AdjudicatorEmail>lindh@nomail.com</AdjudicatorEmail>
  <ClaimantLastName>Smith</ClaimantLastName>
  <OtherLastName>M</OtherLastName>
  <ClaimantFirstName>Ron</ClaimantFirstName>
  <ClaimantMiddleInitial>S</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>
  <FailureToRespondMessage>Failed Message Comments</FailureToRespondMessage>
  <RequestType>1</RequestType>
  <OtherExchangeBRTN>104779533</OtherExchangeBRTN>
  <NumberOfInquiries>3</NumberOfInquiries>
  <InquiryOccurrence>
    <InquiryNumber>01</InquiryNumber>
    <InquiryQuestion>You indicated the claimant was discharge for drugs. What made you
believe the claimant was on drugs</InquiryQuestion>
  </InquiryOccurrence>
  <InquiryOccurrence>
    <InquiryNumber>02</InquiryNumber>
    <InquiryQuestion>Do you have a drug testing policy. If yes provide policy</InquiryQuestion>
  </InquiryOccurrence>
</StateAdditionalFactFindingRequest>

```



```

</InquiryOccurrence>
<InquiryOccurrence>
  <InquiryNumber>03</InquiryNumber>
  <InquiryQuestion>Was the claimant given a drug test. If no why was the claimant not
  tested.If yes what was the result of the drug test</InquiryQuestion>
</InquiryOccurrence>
<EvidenciaryAttachmentOccurrence>
  <DescriptionofAttachmentCode>3</DescriptionofAttachmentCode>
  <TypeofDocument>TypeofDocument1</TypeofDocument>
  <AttachmentSize>1</AttachmentSize>
  <AttachmentData>MA==</AttachmentData>
</EvidenciaryAttachmentOccurrence>
<RequestDate>2019-09-10</RequestDate>
<ResponseDueDate>2019-09-21</ResponseDueDate>
<FormNumber>006B</FormNumber>
</StateAdditionalFactFindingRequest>
</StateAdditionalFactFindingRequestCollection>

```

### 3.2.12 Sample Integrated Partner Additional Fact-Finding Response

**REDACTED**

```

<EmployerTPAAdditionalFactFindingResponse>
  <StateRequestRecordGUID>11000000000000000000000000000001</StateRequestRecordGUID>
  <BrokerRecordTransactionNumber>13156</BrokerRecordTransactionNumber>
  <RequestDate>2019-09-10</RequestDate>
  <ResponseDueDate>2019-09-21</ResponseDueDate>
  <SSN>000110001</SSN>
  <ClaimEffectiveDate>2019-09-10</ClaimEffectiveDate>
  <ClaimNumber>11231</ClaimNumber>
  <StateEmployerAccountNbr>179424876</StateEmployerAccountNbr>
  <EmployerName>Pennies Inc</EmployerName>
  <ClaimantLastName>Smith</ClaimantLastName>
  <ClaimantFirstName>Ron</ClaimantFirstName>
  <ClaimantMiddleInitial>S</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>
  <AdjudicatorIdentifier>ADJ101</AdjudicatorIdentifier>
  <RequestType>1</RequestType>
  <NumberOfInquiries>3</NumberOfInquiries>
  <InquiryOccurrence>
    <InquiryNumber>01</InquiryNumber>
    <InquiryQuestion>You indicated the claimant was discharge for drugs. What made you
    believe the claimant was on drugs</InquiryQuestion>
    <ResponseComments>Claimant Doctor Reports</ResponseComments>
  </InquiryOccurrence>
  <InquiryOccurrence>
    <InquiryNumber>02</InquiryNumber>
    <InquiryQuestion>Do you have a drug testing policy. If yes provide policy</InquiryQuestion>
    <ResponseComments>Yes, Policy handbook attached sent via email on
    request</ResponseComments>
  </InquiryOccurrence>
  <InquiryOccurrence>
    <InquiryNumber>03</InquiryNumber>

```

<InquiryQuestion>Was the claimant given a drug test. If no why was the claimant not tested.If yes what was the result of the drug test</InquiryQuestion>  
<ResponseComments>Yes, claimant was tested before hired</ResponseComments>  
</InquiryOccurrence>

<PreparerTypeCode>E</PreparerTypeCode>  
<PreparerTelephoneNumberPlusExt>333333333333</PreparerTelephoneNumberPlusExt>  
<PreparerContactName>Ronny Melford</PreparerContactName>  
<PreparerTitle>Preparor Level1</PreparerTitle>  
<PreparerEmailAddress>Ronny@asdf.com</PreparerEmailAddress>  
</EmployerTPAAdditionalFactFindingResponse>  
</EmployerTPAAdditionalFactFindingResponseCollection>

### 3.3 XSD

On the client Connector, as well as the Central Broker, the request and response files must validate against the XML schema definition.

In order to implement the XML for SIDES data, the Exchange Formats were translated into XML Schema Definition (XSD) files. These XSD files are used in SIDES to validate the State request and the Integrated Partner response.

Any violation of the XSD will result in an error indicating that the request or response was not successfully processed and must be fixed by the sender and resubmitted to SIDES.

#### 3.3.1 Separation Information XSD

There are three files that make up this definition for Separation Information.

Two main files make up the schema:

- **REDACTED**

One support file contains elements that are defined in both files.

- **REDACTED**

There is one XSD file, **REDACTED**.xsd, which is used to include other XSD files in the system, and it does not contain any additional information. This file is required due to a problem accessing the **REDACTED** namespace in multiple files within the Java libraries used in SIDES. The combinedSI.xsd file is used internally by the Central Broker to allow XSD checks to take place on all the SOAP messages and records sent in by the Connectors. The **REDACTED**.xsd file may be used by connector software, but it is not necessary if the technology and libraries used in the Connectors' implementation do not require it.

- **REDACTED**.xsd

##### 3.3.1.1 XSD Files

**REDACTED**

.

#### 3.3.2 Earnings Verification XSD

There are three files that make up this definition for Earnings Verification.

Two main files make up the schema:

- **REDACTED**

One support file contains elements that are defined in both files.

- **REDACTED**

The **REDACTED**.xsd mentioned above in Separation Information is reused. There is one XSD file, **REDACTED**.xsd, which is used to include other XSD files in the system, and it does not contain any additional information. This file is required due to a problem accessing the **REDACTED** namespace in multiple files within the Java libraries used in SIDES. The combinedEV.xsd file is used internally by the Central Broker to allow XSD checks to take place on all the SOAP messages and records sent in by the Connectors. The **REDACTED**.xsd file may be used by connector software, but it is not necessary if the technology and libraries used in the Connectors' implementation do not require it.

- **REDACTED**.xsd

### 3.3.2.1 XSD Files

**REDACTED**

### 3.3.3 Monetary & Potential Charges XSD

There are three files that make up this definition for Monetary & Potential Charges.

Two main files make up the schema:

- **REDACTED**

One support file contains elements that are defined in both files.

- **REDACTED**

The **REDACTED**.xsd mentioned above in Separation Information is reused.

There is one XSD file, **REDACTED**.xsd, which is used to include other XSD files in the system, and it does not contain any additional information. This file is required due to a problem accessing **REDACTED** namespace in multiple files within the Java libraries used in SIDES. The **REDACTED**.xsd file is used internally by the Central Broker to allow XSD checks to take place on all the SOAP messages and records sent in by the Connectors. The **REDACTED**.xsd file may be used by connector software, but it is not necessary if the technology and libraries used in the Connectors' implementation do not require it.

- **REDACTED**.xsd

### 3.3.3.1 XSD Files

**REDACTED**

### 3.3.4 Determinations & Decisions XSD

There are three files that make up this definition for Determinations & Decisions.

Two main files make up the schema:

- **REDACTED**

One support file contains elements that are defined in both files.

- **REDACTED**

The **REDACTED.xsd** mentioned above in Separation Information is reused.

There is one XSD file, **REDACTED.xsd**, which is used to include other XSD files in the system, and it does not contain any additional information. This file is required due to a problem accessing the **REDACTED** namespace in multiple files within the Java libraries used in SIDES. The **REDACTED.xsd** file is used internally by the Central Broker to allow XSD checks to take place on all the SOAP messages and records sent in by the Connectors. The **REDACTED.xsd** file may be used by connector software, but it is not necessary if the technology and libraries used in the Connectors' implementation do not require it.

- **REDACTED.xsd**

#### 3.3.4.1 XSD Files

**REDACTED**

#### 3.3.5 Benefit Charges XSD

There are three files that make up this definition for Benefit Charges.

Two main files make up the schema:

- **REDACTED**

One support file contains elements that are defined in both files.

- **REDACTED**

The RequestResponseTypeElem **REDACTED.xsd** mentioned above in Separation Information is reused.

There is one XSD file, **REDACTED.xsd**, which is used to include other XSD files in the system, and it does not contain any additional information. This file is required due to a problem accessing the **REDACTED** namespace in multiple files within the Java libraries used in SIDES. The **REDACTED.xsd** file is used internally by the Central Broker to allow XSD checks to take place on all the SOAP messages and records sent in by the Connectors. The **REDACTED.xsd** file may be used by connector software, but it is not necessary if the technology and libraries used in the Connectors' implementation do not require it.

- **REDACTED.xsd**

### 3.3.5.1 XSD Files

**REDACTED**

### 3.3.6 Additional Fact-Finding XSD

There are three files that make up this definition for Additional Fact-Finding.

Two main files make up the schema:

- **REDACTED**

One support file contains elements that are defined in both files.

- **REDACTED**

The **REDACTED**.xsd mentioned above in Separation Information is reused.

There is one XSD file, **REDACTED**.xsd, which is used to include other XSD files in the system, and it does not contain any additional information. This file is required due to a problem accessing the **REDACTED** namespace in multiple files within the Java libraries used in SIDES. The **REDACTED**.xsd file is used internally by the Central Broker to allow XSD checks to take place on all the SOAP messages and records sent in by the Connectors. The combinedAF.xsd file may be used by connector software, but it is not necessary if the technology and libraries used in the Connectors' implementation do not require it.

- **REDACTED**.xsd

### 3.3.6.1 XSD Files

**REDACTED**

### 3.3.7 Null/Empty Values

In many instances, the Exchange Format and the XSD indicate that an element can be null or is not required. There are two main ways to represent null values in XML (strings being a special case):

- One is to include xsi:nil="true" if the element in question is supposed to be null
- The other is to not include the element

For SIDES, the way to indicate null values is to not include the element. Therefore, any element that does not have a value must **not** appear in the XML file sent to the Central Broker. This is to keep the footprint of the record as small as possible on all messages sent to and from the Central Broker. Optional string element values are a special case in that they will successfully pass the xml parser even if they have a null value. The Central Broker checks for null values in the java

code and returns an XSD validation error to the State/Integrated Partner if this case does indeed occur.

The ClaimantFirstName and ClaimantLastName data elements in the Separation Request are required. Yet, there may not be a value that can be placed in them (in the case where the claimant does not have either a first or last name); if this occurs, a space must be sent in as the element value.

### 3.3.8 Dates

The XSD defines all of the Date data types as *xs:date*. This data type allows the definition of a year, month, and a day to define the particular date. The Exchange Format date is restricted to just the year, month and day (10 characters total). It is important to keep the *xs:date* field to just the year, month, and day.

There are two exceptions to this rule where the full date/time is used. This is in the case of the Broker Effective Date field and the DateStartedReceivingTransmission/ DateFinishedReceivingTransmission fields in the message acknowledgements. The Broker Effective Date indicates when a record was received in the Central Broker and requires the use of the date and time fields in *xs:dateTime* to record the exact time of record receptions and transmissions by the Broker. The DateStartedReceivingTransmission/ DateFinishedReceivingTransmission are part of the acknowledgements and give the receiver the date and time box around the transmission. The time zones for each of these elements will always be in Greenwich Mean Time or an offset thereof.

### 3.3.9 MTOM

Message Transmission Optimization Mechanism, or MTOM, is a mechanism for transmitting large binary attachments with SOAP messages as raw bytes, allowing for smaller messages. Binary content often has to be re-encoded to be sent as text data with SOAP messages. MTOM allows more efficient sending of binary data in a SOAP request or response. MTOM provides a way of efficiently transmitting binary data such as images, PDF files, and RTF documents, between Connectors.

The basis of MTOM used is the data type *base64Binary* (<http://www.w3.org/TR/2004/PER-xmlschema-2-20040318/#base64Binary>).

This is defined in the Request and Response XSD as part of the attachment occurrence:

```
<xs:element name="AttachmentData" type="xs:base64Binary" />
```

Connectors must use the *base64Binary* data type for their request/response attachments.

### 3.3.10 Backfilled Data

There are some backfilled data required in the response. This data shall come out of the request record in the exact form it is received. Central Broker business rule checks determine if the data matches between the request and response on these fields. If there is any difference, the Central Broker rejects the record.

The fields that must be backfilled are given in Table 2, 3, 4, 5, and 6.

**Table 2 – Separation Information Backfilled Data**

<b>Element Name</b>	<b>Separation Information Exchange Format Sequence Number</b>	<b>Notes</b>
StateRequestRecordGUID	B-59	
BrokerRecordTransactionNumber	B-60	
SSN	B-1	
ClaimEffectiveDate	B-2	
ClaimNumber	B-3	If this was not included in the request, then it must not be included in the response.
StateEmployerAccountNbr	B-4	
EmployerName	B-65	
ClaimantFirstName	B-67	
ClaimantLastName	B-66	
ClaimantMiddleInitial	B-68	If this was not included in the request, then it must not be included in the response.
ClaimantSuffix	B-69	If this was not included in the request, then it must not be included in the response.



**Table 3 – Earnings Verification Backfilled Data**

Element Name	Earnings Verification Exchange Format Sequence Number	Notes
StateEarningsVerificationRequestRecordGUID	ER-1	
BrokerRecordTransactionNumber	ER-2	
RequestingStateAbbreviation	ER-3	
UIOfficeName	ER-4	
StateEmployerAccountNbr	ER-5	
FEIN	ER-6	
EmployerName	ER-8	
SSN	ER-10	
EarningsVerificationWeekEndDate	ER-13	
EarningsVerificationWeekEndDate	ER-14	
EarningsVerificationSourceCode	ER-32	
ClaimantLastName	ER-34	
ClaimantFirstName	ER-35	
ClaimantMiddleInitial	ER-36	If this was not included in the request, then it must not be included in the response.

Element Name	Earnings Verification Exchange Format Sequence Number	Notes
ClaimantSuffix	ER-37	If this was not included in the request, then it must not be included in the response.
RequestDate	ER-38	
ResponseDueDate	ER-39	

**Table 4 - Monetary & Potential Charges Backfilled Data**

Element Name	Monetary & Potential Charges Exchange Format Sequence Number	Notes
StateMonetaryChargeRequestRecordGUID	MR-1	
BrokerRecordTransactionNumber	MR-2	
ClaimNumber	MR-3	
StateEmployerAccountNbr	MR-5	
SSN	MR-8	
EmployerName	MR-33	
ClaimantLastName	MR-34	

<b>Element Name</b>	<b>Monetary &amp; Potential Charges Exchange Format Sequence Number</b>	<b>Notes</b>
ClaimantFirstName	MR-35	
ClaimantMiddleInitial	MR-36	If this was not included in the request, then it must not be included in the response.
ClaimantSuffix	MR-37	If this was not included in the request, then it must not be included in the response.
RequestDate	MR-38	
ResponseDueDate	MR-39	

**Table 5 - Determinations & Decisions Backfilled Data**

<b>Element Name</b>	<b>Determinations &amp; Decisions Exchange Format Sequence Number</b>	<b>Notes</b>
StateDeterminationDecisionRequestRecordGUID	DR-32	
BrokerRecordTransactionNumber	DR-33	
RequestDate	DR-53	

Element Name	Determinations & Decisions Exchange Format Sequence Number	Notes
LastDateToAppeal	DR-54	
RequestingStateAbbreviation	DR-1	
SSN	DR-2	
ClaimantLastName	DR-3	
OtherLastName	DR-4	If this was not included in the request, then it must not be included in the response.
ClaimantFirstName	DR-5	
ClaimantMiddleInitial	DR-6	If this was not included in the request, then it must not be included in the response.
ClaimantSuffix	DR-7	If this was not included

Element Name	Determinations & Decisions Exchange Format Sequence Number	Notes
		in the request, then it must not be included in the response.
ClaimEffectiveDate	DR-8	
ClaimNumber	DR-9	If this was not included in the request, then it must not be included in the response.
BenefitYearBeginDate	DR-10	
StateEmployerAccountNbr	DR-11	
FEIN	DR-12	
EmployerName	DR-13	
DeterminationOrDecisionInd	DR-14	
DecisionNumberOrDocketNumber	DR-15	If this was not included in the request,

Element Name	Determinations & Decisions Exchange Format Sequence Number	Notes
		then it must not be included in the response.
DeterminationOrDecisionDate	DR-16	

**Table 6 – Benefit Charges Backfilled Data**

Element Name	Benefit Charge Exchange Format Sequence Number	Notes
StateChargeRecordGUID	CR-32	
BrokerRecordTransactionNumber	CR-33	
NoticeDate	CR-1	
LastDayToRespond	CR-2	
ChargingStateAbbreviation	CR-3	
StateEmployerAccountNbr	CR-4	
EmployerName	CR-5	
FEIN	CR-6	
ChargingStatementStartDate	CR-7	
ChargingStatementEndDate	CR-8	

Element Name	Benefit Charge Exchange Format Sequence Number	Notes
SSN	CR9.1	
ClaimNumber	CR-9.2	
ClaimantLastName	CR-9.3	
OtherLastName	CR-9.4	If this was not included in the request, then it must not be included in the response.
ClaimantFirstName	CR-9.5	
ClaimantMiddleInitial	CR-9.6	If this was not included in the request, then it must not be included in the response.
ClaimantSuffix	CR-9.7	If this was not included in the request, then it must not be included in the response.
BenefitYearBeginDate	CR-9.8	
ProgramCode	CR-9.9	
TypeofEmployerCode	CR-9.9	

**Table 7 – Additional Fact-Finding Backfilled Data**

Element Name	Additional Fact-Finding Exchange Format Sequence Number	Notes
StateChargeRecordGUID	ADFR-1	
BrokerRecordTransactionNumber	ADFR -2	
RequestDate	ADFR -3	
ResponseDueDate	ADFR -4	
SSN	ADFR -5	
ClaimEffectiveDate	ADFR -6	
ClaimNumber	ADFR -7	
StateEmployerAccountNbr	ADFR -8	
ClaimantLastName	ADFR -15	
ClaimantFirstName	ADFR -16	
ClaimantMiddleInitial	ADFR -17	If this was not included in the request, then it must not be included in the response.
ClaimantSuffix	ADFR -18	If this was not included in the request, then it must not be included in the response.
AdjudicatorIdentifier	ADFR -19	
NumberOfInquires	ADFR -21	



Element Name	Additional Fact-Finding Exchange Format Sequence Number	Notes
InquiryNumber	ADFR -22.1	
UnquiryQuestion	ADFR-22.2	

### 3.4 File Size

States may include multiple requests in a single XML file per Integrated Partner to which they want to send requests. This file may be up to 8MB in size, including all encoded attachments, prior to encryption. If States have more than 8MB of data for that Integrated Partner, they must create multiple files.

Similarly, multiple Integrated Partner responses to a State may be packaged into a single file within the 8MB limit. If Integrated Partners have more than 8MB of data for a State (including encoded attachments but prior to encryption), they must also create multiple files.

The 8MB limit on file size implies an 8MB limit on the individual record size also. Records cannot span multiple files.

### 3.5 Business Rules

The Business Rules column in the Exchange Formats contains additional business rule validation logic (“edits”) that cannot be defined in an XSD. To keep the data in SIDES consistent, the Central Broker uses Java code to check every request and response record that is sent to it. The Central Broker verifies that all of the business rules are followed. If it detects any of these rules have been violated, it passes back an error code and message to the calling client that indicates the rules that were violated.

An example of a Java implemented business rule (rather than a validation that can be implemented within the XSD) is given below for the Separation Information exchange. This example logs error 2114 (see Part B – Connector Requirements) if the associated business rule is violated.

```

if (res.getFraudReasons() == null && IndicatorTypeY.Y.equals(res.getFraudInd())) {
    errorList.add(new BRValidationError(2114));
    logger.debug("Found error (2114) - Business Rule violation - FraudReasons (B-216)
required if FraudInd (B-225) is Y for Yes.");
}

```

In order to create a connector, the software must implement and execute all of the validations and business rules specified in the Exchange Formats prior to sending the request or response to the Broker.

## 4. BUILD THE CONNECTOR: PREPARING THE MESSAGE

Prior to transmittal of the request or response data to the Central Broker, the Connector must extract the data from its backend system and generate an XML file. Once the XML file has been created, the file must be packaged and delivered to the Central Broker. This section describes the messaging framework that a Connector can use to deliver messages to and receive messages from the Central Broker.

**Note:** The NASWA SIDES team added an additional file type that records can be constructed in if the Connector uses the Model Connector. This file type is an ASCII file type. Even though the Connector creates its files in this type, it gets converted into XML before being sent. Therefore, all discussion with the file types is in XML.

### 4.1 Messaging Overview – Post and Pull

The communication between the Central Broker, the State and Integrated Partner is accomplished through **REDACTED** over **REDACTED** using an **REDACTED** pattern.

There are two operations supported by the Central Broker – “Post” and “Pull.”

#### 4.1.1 Post

In the Post operation, the connecting client instigates communication with the Broker. The Connectors “Post” their request and response files to the Central Broker (**REDACTED**) and receive an acknowledgement in return (**REDACTED**).

Post Transaction:

1. Connector posts its request file (if a State), or response file (if an Integrated Partner) to the Broker in an **REDACTED**
2. Connector receives receipt of file in an **REDACTED** from the Broker

#### 4.1.2 Pull

In the Pull operation, the Connector asks for any available records from the Central Broker (**REDACTED**), receives the waiting records (**REDACTED**) and sends back an acknowledgement in return (**REDACTED**). Because there are three communications that take place with this action, the process is broken up into **REDACTED** distinct **REDACTED** transactions:

Pull Transaction (1):

1. Connector asks for its files (responses if a State, requests if an Integrated Partner) in an **REDACTED** to the Broker
2. Connector receives its files in an **REDACTED** from the Broker

Pull Transaction (2):

3. Connector **REDACTED** of file in an **REDACTED** to the Broker

**Note: REDACTED**

### 4.1.3 SOAP

SOAP is a protocol for exchange of information in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts: (1) an envelope that defines a framework for describing what is in a message and how to process it, (2) a set of encoding rules for expressing instances of application-defined data types, and (3) a convention for representing remote procedure calls and responses.

SOAP messages are fundamentally one-way transmissions from a sender to a receiver. But SOAP messages are often combined to implement patterns such as the request/response pattern, where it provides for SOAP response messages to be delivered as HTTP responses, using the same connection as the inbound request. This is the pattern used to accomplish the file **REDACTED** scheme, as overviewed in Sections 4.1.1, 4.1.2, and **Error! Reference source not found..**

## 4.2 Messaging Concepts

There are a few concepts that need to be discussed before getting further into the messaging process for SIDES.

### 4.2.1 Unique ID

Each State and Integrated Partner will be assigned a Unique ID for both the SIDES production environment and the SIDES test environment. Please contact the NASWA SIDES team at [uisides@naswa.org](mailto:uisides@naswa.org) to obtain your Unique IDs for SIDES. A complete list of Unique IDs is maintained on the SIDES Website (<https://info.uisides.org>). The Unique IDs of all the current participants are as follows:

#### 4.2.1.1 State Unique IDs

Table 8 lists the participating States and their Unique IDs:

**Table 8 - Unique IDs of Current Participating States**

State	Unique ID
<b>REDACTED</b>	

State	Unique ID

#### 4.2.1.2 Integrated Partner Unique IDs

Table 9 lists the participating Integrated Partners and their Unique IDs. The Integrated Partner Unique ID is a 'BR' followed by a unique nine digit number. The NASWA SIDES team assigns the nine digit number.

**Table 9 - Unique IDs of Current Participating Integrated Partners**

Integrated Partner	Unique ID
<b>REDACTED</b>	

#### 4.2.2 File and Record GUIDs

SIDES uses a Globally Unique Identifier (GUID). A GUID is a special type of identifier used in software applications to provide a unique reference number. The value is represented as a **REDACTED** character hexadecimal character string, such as { **REDACTED** }. The primary purpose of the GUID is to have a totally unique number. Ideally, a GUID will never be generated twice by any computer or group of computers in existence. The total number of unique keys (**REDACTED**) is so large that the probability of the same number being generated twice is extremely small. The NASWA SIDES team suggests GUIDs be generated using utilities or function calls available within your development framework.

**NOTE:** States must not reuse a GUID unless it is was used more than 10 years ago. GUIDs may only be reused for a resend of a request that was rejected by the **Central Broker**.

GUIDs are created by Connectors for each "Post" transaction to the Broker and are used on both the XML file level (File GUIDs) and the XML individual record level (Record GUIDs).

The File GUIDs are used on an entire XML file to uniquely identify that file.

The Record GUIDs are used within the XML on each record to uniquely identify that record. Record GUIDs should be unique with the domain of the Connector and not just that file.

There are many GUID creation methods depending on the language and technology in use. Many of them will create the GUID with a **REDACTED** in a few places such that it actually comes out to be **REDACTED** characters long. **REDACTED**.

**NOTE:** Because of the small chance that a GUID will be repeated by different states, employer/TPAs must ensure that both the State Request Record GUID **AND** the State abbreviation (or something similar) be used when determining if a record is a duplicate.

#### 4.2.3 SOAP Transaction Number

The Central Broker uses a SOAP transaction number (StateSOAPTransactionNumber and EmployerTPASOAPTransactionNumber) as a unique identifier for a file as part of a “Pull” transaction. These can be thought of like a FedEx or UPS tracking number. These numbers can be used in calls to re-Pull a particular file in case of loss.

#### 4.2.4 Broker Record Transaction Number

The BrokerRecordTransactionNumber is given to each individual request record on entry into the system and is generated by the Broker. This number uniquely identifies a request, even if multiple copies of the same record are passed through the system (in this case, each record gets its own BrokerRecordTransactionNumber). The BrokerRecordTransactionNumber must be used on the response in order to connect the response with a particular request and, therefore, must be consumed by Integrated Partner Connectors.

#### 4.2.5 Message Codes

Each time an acknowledgement is sent, a message code is sent with it to indicate the status of the transaction. These are the transmission-level message codes that go in the SOAP header as discussed below in Section 4.3 SOAP Custom Headers.

##### 4.2.5.1 Post-Acknowledgement Message Codes

**Table 10 – Post-Acknowledgement Message Codes**

Code	Message	Notes
1	File Success	Successful Transmission; no rejects

Code	Message	Notes
2	File Failure	File size too large; no records in file; all records failed
3	File Success with Rejected Records	Rejected records included

#### 4.2.5.2 Pull-Response Message Codes

**Table 11 – Pull-Response Message Codes**

Code	Message	Notes
1	File Contained in Payload	The file is contained in the payload of the SOAP Message
2	End Of Files	There are no files available to download

#### 4.2.5.3 Pull Acknowledgement Codes

**Table 12 - Pull Acknowledgement Codes**

Code	Message	Notes
1	File Success	Successful transmission
2	File Failure	Did not receive file; any file problems

### 4.3 SOAP Custom Headers

This section discusses the contents of the custom SOAP headers of the various transactions that occur in the Post, Pull, and Push processes.

The SOAP headers are one part of a SOAP message. The SOAP messages contain additional custom information in the SOAP headers. This information utilizes the messaging concepts discussed in the previous Section 4.2 Messaging Concepts. Each type of transaction has its own requirements and elements as discussed below.

The custom SOAP header information provided by the connecting client to the Broker is for routing purposes, security purposes and, in the case of the Pull, the type of Pull required.

The custom SOAP header information provided by the Broker to the client is for security purposes and, in the case of the Pull, for re-Pulling purposes.

The SOAP headers provided for each type of transaction are listed below.

- State Post (Section 4.3.1)
  - State Post to Broker (for routing to specified Integrated Partner)
  - State Post to Broker (for routing to the SIDES E-Response site)
  - Broker Acknowledgement to State Post
- State Pull from Broker – Regular Pull (Section 4.3.2.1)
  - State Request to Broker (Regular Pull)
  - Broker Response to Request (Regular Pull)
  - State Acknowledgement to Broker (Regular Pull)
- State Pull from Broker – Re-Pull by StateSOAPTransactionNumber (Section 4.3.2.2)
  - State Request to Broker (Re-Pull by StateSOAPTransactionNumber)
  - Broker Response to Request (Re-Pull by StateSOAPTransactionNumber)
  - State Acknowledgement to Broker (Re-Pull by StateSOAPTransactionNumber)
- State Pull from Broker – Re-Pull by Date (Section 4.3.2.3)
  - State Request to Broker (Re-Pull by Date)
  - Broker Response to Request (Re-Pull by Date)
  - State Acknowledgement to Broker (Re-Pull by Date)
- Integrated Partner Post (Section 4.3.3)
  - Integrated Partner Post to Broker
  - Broker Acknowledgement to Integrated Partner Post
- Integrated Partner Pull from Broker – Regular Pull (Section 4.3.4.1)
  - Integrated Partner Request to Broker (Regular Pull)
  - Broker Response to Request (Regular Pull)
  - Integrated Partner Acknowledgement to Broker (Regular Pull)



- Integrated Partner Pull from Broker – Re-Pull by EmployerTPASOAPTransactionNumber (Section 4.3.4.2)
  - Integrated Partner Request to Broker (Re-Pull by EmployerTPASOAPTransactionNumber)
  - Broker Response to Request (Re-Pull by EmployerTPASOAPTransactionNumber)
  - Integrated Partner Acknowledgement to Broker (Re-Pull by EmployerTPASOAPTransactionNumber)
- Integrated Partner Pull from Broker – Re-Pull by Date (Section 4.3.4.3)
  - Integrated Partner Request to Broker (Re-Pull by Date)
  - Broker Response to Request (Re-Pull by Date)
  - Integrated Partner Acknowledgement to Broker (Re-Pull by Date)

In the sections that follow, the tables that describe messages going to the Broker have a column that indicates which fields are required. Tables that describe messages being returned from the Broker do not have this column, as there is no responsibility on the client Connector to populate these fields. The client Connector must handle whatever is returned by the Broker according to the header specification.

### 4.3.1 State Post

The following sections contain the custom header elements for a State Post to the Broker.

#### 4.3.1.1 State Post to Central Broker

**Table 13 - State Post to Broker**

Header Element	Required	Definition	Example
To	Y	The Unique ID of the Integrated Partner to which the message is intended  Will always be 'BR' followed by nine digits	BR000000003
From	Y	The Unique ID of the State where the message originated	UT
StateRequestFile GUID	Y	The state-generated GUID applied to this message that can uniquely identify this file	A42A1FBDAC954 9AC7D8D3F45E4 040319

Header Element	Required	Definition	Example
		Size is 32 hexadecimal digits	

#### 4.3.1.1.1 SOAP Example - State Post to Central Broker:

**REDACTED**

#### 4.3.1.2 State Post to Central Broker – SIDES Employer Website (E-Response)

There are two ways to Post to E-Reponse. The first is the original way where you change the TO to a 9-digit FEIN, include a SEIN and PIN and send an individual message to every employer.

**Table 14 - State Post to Broker - SIDES Employer Website (E-Response)**

Header Element	Required	Definition	Example
To	Y	The FEIN of the employer or TPA to which the message is intended Size is nine numeric digits	123456789
From	Y	The Unique ID of the State where the message originated	NJ
StateRequestFile GUID	Y	The state-generated GUID applied to this message that can uniquely identify this file Size is 32 hexadecimal digits	A42A1FBDAC954 9AC7D8D3F45E4 040319
SEIN	Y	The SEIN of the employer or TPA to which the message is intended. For those States that do not use the SEIN, this must equal the FEIN Size is up to 20 digits	123456789
PIN	Y	The PIN to which the State wants to assign this request for this employer or TPA Size is up to 20 characters	435222169876

#### 4.3.1.2.1 SOAP Example - State Post to Central Broker – SIDES Employer Website (E-Response):

**REDACTED**

#### 4.3.1.2.2 State Post to Central Broker – SIDES Employer Website (E-Response) Bundle

There is a second way to post records to E-Response. When creating records that go to E-Response, an additional element has been added to the standard format that will allow the bundling of records into a single file. The FEIN and State Employer Account Number (SEIN) are already required on the record in the exchange. The new element added to the request record standard format is the PIN element. This is the PIN number that the FEIN/SEIN uses to log onto the E-Response website. Therefore, it is now possible to bundle requests to E-Response into a single file instead of having possibly hundreds or thousands of individual requests.

All the same standards still apply on the record, file and SOAP message. For example, if the file size is greater than 8 MB then it will need to be split into two files.

The 'To' in the SOAP headers must be 'EResponse' in order to utilize this method. If the Central Broker receives a record with a PIN value in that record, but does not have 'EResponse' as the 'To,' an error is sent back to the sender. Likewise, if the Central Broker receives a record without the PIN value set in the record and the 'To' is EResponse, an error is passed back to the sender.

The way an E-Response record is sent in to the Central Broker is the same way it will be received on the Pull. So, if a record is posted as part of an E-Response bundle, it will be pulled in an E-Response bundle (not necessarily the same E-Response bundle depending on what responses are waiting to be pulled). The SOAP message will be labeled as 'From: EResponse' in the SOAP headers. If a record is sent into the E-Response as an individual request with the SOAP headers defining the FEIN, SEIN and PIN, then it will be pulled from that individual employer (so the State may have to call for the Pull hundreds or thousands of times depending on how many records the State has waiting to be pulled).

**Note:** If the Model Connector is in use and bundling is the desired method of inserting records into the E-Response, the 'createPin' parameter in the configuration file no longer applies. The Model Connector will not change nor add an element in a request record. Therefore, even if the createPin parameter is set to true, a new PIN will NOT be created inside the record. The original way of inserting records into E-Response with the FEIN, SEIN and PIN in the SOAP headers must be used if this is the desired method of PIN creation.

**Table 15 – State Post to Broker – SIDES Employer Website (E-Response)**

Header Element	Required	Definition	Example
To	Y	Must be 'Eresponse'	Eresponse
From	Y	The Unique ID of the State where the message originated	NJ
StateRequestFile GUID	Y	The state-generated GUID applied to this message that can uniquely identify this file  Size is 32 hexadecimal digits	A42A1FBDAC954 9AC7D8D3F45E4 040319

**4.3.1.2.2.1 SOAP Example – State Post to Central Broker – SIDES Employer Website (E-Response):**

**REDACTED**

**4.3.1.3 Central Broker Acknowledgement to State Post**

**Table 16 – Broker Acknowledgement to State Post**

Header Element	Definition	Example
To	The Unique ID of the State that sent the Post	UT
From	Will always be "Broker"	Broker
StateRequestFileGUID	The state-generated GUID applied to the message that uniquely identifies the file sent in to the Broker. This is for verification purposes  Size is 32 hexadecimal digits	A42A1FBDAC954 9AC7D8D3F45E4 040319
MessageCode	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes.  Size is one digit	1

#### 4.3.1.3.1 SOAP Example - Central Broker Acknowledgement to State Post:

**REDACTED**

#### 4.3.2 State Pull

Because of the nature of the HTTP request-response pattern, there is one request and one response for each HTTP request-response transaction. Because three messages are sent between the Connector and the Broker on a Pull, there will be two request-response patterns needed to accomplish the full Pull operation. (See Section 4.1.2 for an overview of this issue.)

##### 4.3.2.1 State Pull from Central Broker – Regular Pull

The following sections contain the custom header elements for a regular State Pull from the Broker.

##### 4.3.2.1.1 State Request to Central Broker (Regular Pull)

**Table 17 - State Request to Broker (Regular Pull)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the State where this message originated	UT
PullCollection	Y	Signifies one of three Pull transactions desired by the State 1 - indicates a regular Pull Size is one digit	1

##### 4.3.2.1.1.1 SOAP Example - State Request to Central Broker (Regular Pull):

**REDACTED**

##### 4.3.2.1.2 Central Broker Response to Request (Regular Pull)

**Table 18 - Broker Response to Request (Regular Pull)**

Header Element	Definition	Example
To	The Unique ID of the State that requested the Pull	UT
From	The Unique ID of the Integrated Partner from which these response records originated	BR000000001
StateSOAPTransaction Number	The unique number assigned to this file by the Broker	3565
MessageCode	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes. Size is one digit	1

**4.3.2.1.2.1 SOAP Example – Central Broker Response to Request (Regular Pull):**

**REDACTED**

**4.3.2.1.3 State Acknowledgment to Central Broker (Regular Pull)**

**Table 19 - State Acknowledgement to Broker (Regular Pull)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the State from which this message originated	UT
StateSOAPTransaction Number	Y	The StateSOAPTransaction Number that was returned in the response for the regular pull	3565
MessageCode	Y	The acknowledgement code applied to the	1

Header Element	Required	Definition	Example
		message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes. Size is one digit	

**4.3.2.1.3.1 SOAP Example – State Acknowledgment to Central Broker (Regular Pull):**

**REDACTED**

**4.3.2.2 State Pull from Central Broker – Re-Pull by StateSOAPTransactionNumber**

The following sections contain the custom header elements for a State Re-Pull by StateSOAPTransactionNumber from the Broker.

**4.3.2.2.1 State Request to Central Broker (Re-Pull by StateSOAPTransactionNumber)**

**Table 20 - State Request to Broker (Re-Pull by StateSOAPTransactionNumber)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the State from which this message originated	UT
PullCollection	Y	Signifies one of three Pull transactions desired by the State  2 - Indicates a re-Pull by StateSOAPTransactionNumber  Size is one digit	2

Header Element	Required	Definition	Example
StateSOAPTransactionNumber	Y	The StateSOAPTransactionNumber that was returned in the response for the regular pull on a previous Pull request. This specifies the file the State wants to re-Pull	3565

**4.3.2.2.1.1 SOAP Example - State Request to Central Broker (Re-Pull by StateSOAPTransactionNumber):**

**REDACTED**

**4.3.2.2.2 Central Broker Response to Request (Re-Pull by StateSOAPTransactionNumber)**

**Table 21 - Broker Response to Request (Re-Pull by StateSOAPTransactionNumber)**

Header Element	Definition	Example
To	The Unique ID of the State from which this message originated	UT
From	The Unique ID of the Integrated Partner from which these response records originated	BR000000001
StateSOAPTransactionNumber	The unique number assigned to this file by the Broker . Note: This number will not be same number that was sent in the Re-Pull as each transaction gets it own StateSOAPTransactionNumber for auditing purposes.	3565
MessageCode	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes.	1



Header Element	Definition	Example
	Size is one digit	

**4.3.2.2.2.1 SOAP Example - Central Broker Response to Request (Re-Pull by StateSOAPTransactionNumber):**

**REDACTED**

**4.3.2.2.3 State Acknowledgment to Central Broker (Re-Pull by StateSOAPTransactionNumber)**

**Table 22 -State Acknowledgment to Broker (Re-Pull by StateSOAPTransactionNumber)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the State from which this message originated	UT
StateSOAPTransactionNumber	Y	The StateSOAPTransactionNumber that was returned in the response for the regular pull.	3566
MessageCode	Y	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes. Size is one digit	1

**4.3.2.2.3.1 SOAP Example - State Acknowledgment to Central Broker (Re-Pull by StateSOAPTransactionNumber):**

**REDACTED**

**4.3.2.3 State Pull from Central Broker – Re-Pull by Date**

The following sections contain the custom header elements for a State Re-Pull by Date from the Broker.

**4.3.2.3.1 State Request to Central Broker (Re-Pull by Date)**

**Table 23 - State Request to Broker (Re-Pull by Date)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the State from which this message originated	UT
PullCollection	Y	Signifies one of three Pull transactions desired by the State  3 - Indicates a re-Pull by Date  Size is one digit	3
StateSOAPTransactionNumber	Y	This must not be included for the first call by Date. On subsequent calls, this must be filled in with the NextStateSOAPTransaction Number returned on the previous call in order to collect all records on the date specified	

**4.3.2.3.1.1 SOAP Example - State Request to Central Broker (Re-Pull by Date):**

**REDACTED**

**4.3.2.3.2 Central Broker Response to Request (Re-Pull by Date)**

**Table 24 - Broker Response to Request (Re-Pull by Date)**

Header Element	Definition	Example
To	Broker	Broker
From	The Unique ID of the State from which this message originated	UT
StateSOAPTransactionNumber	The first StateSOAPTransactionNumber that was returned in the response for the regular pull on that date	3565
MessageCode	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes. Size is one digit	1
NextStateSoapTransactionNumber	The next StateSOAPTransactionNumber that was returned in the response for the regular pull on that date. This header will not be included in the response when the last file in the data range is being returned, indicating there are no more files to be sent.	3566

**4.3.2.3.2.1 SOAP Example - Central Broker Response to Request (Re-Pull by Date):**

**REDACTED**

#### 4.3.2.3.3 State Acknowledgment to Central Broker (Re-Pull by Date)

**Table 25 - State Acknowledgement to Broker (Re-Pull by Date)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the State from which this message originated	UT
StateSOAPTransaction Number	Y	The StateSOAPTransactionNumber that was returned in the response for the regular pull	3565
MessageCode	Y	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes. Size is one digit.	1

#### 4.3.2.3.3.1 SOAP Example - State Acknowledgment to Central Broker (Re-Pull by Date)

**REDACTED**

### 4.3.3 Integrated Partner Post

The following sections contain the custom header elements for an Integrated Partner Post to the Broker.

#### 4.3.3.1 Integrated Partner Post to Central Broker

**Table 26 – Integrated Partner Post to Broker**

Header Element	Required	Definition	Example
To	Y	The Unique ID of the State to which this message is intended	UT

Header Element	Required	Definition	Example
From	Y	The Unique ID of the Integrated Partner from which the message originated  Will always be 'BR' followed by nine digits	BR000000003
EmployerTPAResponseFileGUID	Y	The Integrated Partner generated GUID applied to this message that can uniquely identify this file  Size is 32 hexadecimal digits	A42A1FBDAC9549AC7D8D3F45E4040319

#### 4.3.3.1.1 SOAP Example – Integrated Partner Post to Central Broker:

**REDACTED**

#### 4.3.3.2 Central Broker Acknowledgement to Integrated Partner Post

**Table 27 - Broker Acknowledgement to Integrated Partner Post**

Header Element	Definition	Example
To	The Unique ID of the Integrated Partner that sent the Post	BR000000003
From	Will always be "Broker."	Broker
EmployerTPAResponseFileGUID	The Integrated Partner generated GUID applied to the message that uniquely identifies the file sent to the Broker. This is for verification purposes.  Size is 32 hexadecimal digits	A42A1FBDAC9549AC7D8D3F45E4040319
MessageCode	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes.	1

Header Element	Definition	Example
	Size is one digit	

#### 4.3.3.2.1 SOAP Example - Central Broker Acknowledgement to Integrated Partner Post:

**REDACTED**

#### 4.3.4 Integrated Partner Pull

Because of the nature of the HTTP request-response pattern, there is one request and one response for each HTTP request-response transaction. Because three messages are sent between the Connector and the Broker on a Pull, there will be two request-response patterns needed to accomplish the full Pull operation. (See Section 4.1.2 for an overview of this issue.)

#### 4.3.4.1 Integrated Partner Pull from Central Broker – Regular Pull

The following sections contain the custom header elements for a regular State Pull from the Broker.

##### 4.3.4.1.1 Integrated Partner Request to Central Broker (Regular Pull)

**Table 28 – Integrated Partner Request to Broker (Regular Pull)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the Integrated Partner from which this message originated	BR000000003
PullCollection	Y	Signifies one of three Pull transactions desired by the Integrated Partner  1 - Indicates a regular Pull  Size is one digit	1

**4.3.4.1.1 SOAP Example – Integrated Partner Request to Central Broker (Regular Pull):**

**REDACTED**

**4.3.4.1.2 Central Broker Response to Request (Regular Pull)**

**Table 29 - Broker Response to Request (Regular Pull)**

Header Element	Definition	Example
To	The Unique ID of the Integrated Partner that requested the Pull	BR000000003
From	The Unique ID of the State from which these request records originated.	UT
EmployerTPASOAPTransactionNumber	The unique number assigned to this file by the Broker	7350
MessageCode	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes. Size is one digit	1

**4.3.4.1.2.1 SOAP Example – Central Broker Response to Request (Regular Pull):**

**REDACTED**

**4.3.4.1.3 Integrated Partner Acknowledgment to Central Broker (Regular Pull)**

**Table 30 – Integrated Partner Acknowledgment to Broker (Regular Pull)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the Integrated Partner from which this message originated	BR000000003

Header Element	Required	Definition	Example
EmployerTPASOAPTransactionNumber	Y	The EmployerTPASOAPTransactionNumber that was returned in the response for the regular Pull	7350
MessageCode	Y	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes.  Size is one digit	1

#### 4.3.4.1.3.1 SOAP Example – Integrated Partner Acknowledgment to Central Broker (Regular Pull):

**REDACTED**

#### 4.3.4.2 Integrated Partner Pull from Central Broker – Re-Pull by EmployerTPASOAPTransactionNumber

The following sections contain the custom header elements for an Integrated Partner Re-Pull by EmployerTPASOAPTransactionNumber from the Broker.

##### 4.3.4.2.1 Integrated Partner Request to Central Broker (Re-Pull by EmployerTPASOAPTransactionNumber)

**Table 31 – Integrated Partner Request to Broker (Re-Pull by EmployerTPASOAPTransactionNumber)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the Integrated Partner from which this message originated	BR000000001



Header Element	Required	Definition	Example
PullCollection	Y	Signifies one of three Pull transactions desired by the Integrated Partner  2 - Indicates a re-Pull by EmployerTPASOAPTransaction Number  Size is one digit	2
EmployerTPASOAPTransactionNumber	Y	The EmployerTPASOAPTransaction Number that was returned in the response for the regular Pull on a previous Pull request. This specifies the file the Integrated Partner wants to re-Pull	7350

**4.3.4.2.1.1 SOAP Example – Integrated Partner Request to Central Broker (Re-Pull by EmployerTPASOAPTransactionNumber):**

**REDACTED**

**4.3.4.2.2 Central Broker Response to Request (Re-Pull by EmployerTPASOAPTransactionNumber)**

**Table 32 - Broker Response to Request (Re-Pull by EmployerTPASOAPTransactionNumber)**

Header Element	Definition	Example
To	The Unique ID of the Integrated Partner from which this message originated	BR000000001
From	The Unique ID of the Integrated Partner from which these response records originated	UT

Header Element	Definition	Example
EmployerTPASOAPTransactionNumber	<p>The EmployerTPASOAPTransactionNumber that was requested by the Integrated Partner</p> <p>The unique number assigned to this file by the Broker. Note: This number will not be same number that was sent in the Re-Pull as each transaction gets it own EmployerTPASOAPTransactionNumber for auditing purposes.</p>	7351
MessageCode	<p>The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes.</p> <p>Size is one digit.</p>	1

**4.3.4.2.2.1 SOAP Example - Central Broker Response to Request (Re-Pull by EmployerTPASOAPTransactionNumber):**

**REDACTED**

**4.3.4.2.3 Integrated Partner Acknowledgment to Central Broker (Re-Pull by EmployerTPASOAPTransactionNumber)**

**Table 33 – Integrated Partner Acknowledgment to Broker (Re-Pull by EmployerTPASOAPTransactionNumber)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the Integrated Partner from which this message originated.	BR000000001

Header Element	Required	Definition	Example
EmployerTPASOAPTransactionNumber	Y	The EmployerTPASOAPTransactionNumber that was returned in the response for the regular Pull	7351
MessageCode	Y	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes. Size is one digit	1

#### 4.3.4.2.3.1 SOAP Example – Integrated Partner Acknowledgment to Central Broker (Re-Pull by EmployerTPASOAPTransactionNumber):

**REDACTED**

#### 4.3.4.3 Integrated Partner Pull from Central Broker – Re-Pull by Date

The following sections contain the custom header elements for an Integrated Partner Re-Pull by Date from the Broker.

##### 4.3.4.3.1 Integrated Partner Request to Central Broker (Re-Pull by Date)

The first time this operation is called, the EmployerTPASOAPTransactionNumber is null and the dates from which the Connector wants to Re-Pull are included in the SOAP payload.

When the Broker replies with the first file, the Broker will include the next EmployerTPASOAPTransactionNumber during that date range in a SOAP header attribute (NextEmployerTPASOAPTransactionNumber).

In the next call to this operation, the caller includes this EmployerTPASOAPTransactionNumber with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

The last file sent back to the Connector is indicated by a null value for the next EmployerTPASOAPTransactionNumber.

**Table 34 – Integrated Partner Request to Broker (Re-Pull by Date)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the Integrated Partner from which this message originated.	BR000000001
PullCollection	Y	Signifies one of three Pull transactions desired by the Integrated Partner  3 - Indicates a re-Pull by Date  Size is one digit	3
EmployerTPASOAPTransactionNumber	Y	This must not be included for the first call by Date. On subsequent calls, this must be filled in with the NextEmployerTPASOAPTransactionNumber returned on the previous call in order to collect all records on the date specified	

**4.3.4.3.1.1 SOAP Example – Integrated Partner Request to Central Broker (Re-Pull by Date):**

**REDACTED**

**4.3.4.3.2 Central Broker Response to Request (Re-Pull by Date)**

**Table 35 - Broker Response to Request (Re-Pull by Date)**

Header Element	Definition	Example
To	Broker	Broker
From	The Unique ID of the Integrated Partner from which this message originated.	BR000000001

Header Element	Definition	Example
EmployerTPASOAPTransactionNumber	The first EmployerTPASOAPTransactionNumber that was returned in the response for the regular Pull on that date	7350
MessageCode	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes.  Size is one digit	1
NextEmployerTPASOAPTransactionNumber	The next EmployerTPASOAPTransactionNumber that was returned in the response for the regular Pull on that date. This header will not be included in the response when the last file in the data range is being returned, indicating there are no more files to be sent.	7351

**4.3.4.3.2.1 SOAP Example - Central Broker Response to Request (Re-Pull by Date):**

**REDACTED**

**4.3.4.3.3 Integrated Partner Acknowledgment to Central Broker (Re-Pull by Date)**

**Table 36 – Integrated Partner Acknowledgment to Broker (Re-Pull by Date)**

Header Element	Required	Definition	Example
To	Y	Broker	Broker
From	Y	The Unique ID of the Integrated Partner from which this message originated.	BR000000001

Header Element	Required	Definition	Example
EmployerTPASOAPTransactionNumber	Y	The EmployerTPASOAPTransaction Number that was returned in the response for the regular Pull	7350
MessageCode	Y	The acknowledgement code applied to the message that indicates success or failure of the entire transmission. See 4.2.5 for further information on Message Codes. Size is one digit	1

#### 4.3.4.3.3.1 SOAP Example – Integrated Partner Acknowledgment to Central Broker (Re-Pull by Date)

**REDACTED**

### 4.4 SOAP Payload

#### 4.4.1 Separation Information

##### 4.4.1.1 Post Payload

A “Post” is defined (see Section 4.1.1) as sending a request or response to the Broker by a particular Connector. This section discusses the (pre-encryption) payloads in the SOAP message.

##### 4.4.1.1.1 State Post Payload

##### 4.4.1.1.1.1 Post to Central Broker Payload

The “Post” payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED** xsd.

**REDACTED**

```

<StateSeparationRequest>
  <StateRequestRecordGUID>11000000000000000000000000000001</StateRequestRecord
GUID>
  <!--State Info-->
  <RequestingStateAbbreviation>VA</RequestingStateAbbreviation>
  <UIOfficeName>BU1</UIOfficeName>
  <UIOfficePhone>5553247787</UIOfficePhone>

```

```

<UIOfficeFax>5553247787</UIOfficeFax>

<UIOfficeEmailAddress>customerservice@vec.virginia.gov</UIOfficeEmailAddress>
  <AllocationRequiredInd>Y</AllocationRequiredInd>

  <!--Employer Information-->
  <StateEmployerAccountNbr>12000120001200012000</StateEmployerAccountNbr>
  <EmployerName>FedPoint Systems LLC</EmployerName>
  <FEIN>123456789</FEIN>
  <TypeofEmployerCode>3</TypeofEmployerCode>

  <!--Claimant Information-->
  <SSN>123456666</SSN>
  <ClaimantLastName>WWisWO_AllocIndisY</ClaimantLastName>
  <OtherLastName>Bugs</OtherLastName>
  <ClaimantFirstName>AttachInds1</ClaimantFirstName>
  <ClaimantSuffix>Jr</ClaimantSuffix>

  <!--Claim Data-->
  <ClaimEffectiveDate>2021-07-30</ClaimEffectiveDate>
  <TypeofClaimCode>3</TypeofClaimCode>
  <BenefitYearBeginDate>2021-06-04</BenefitYearBeginDate>
  <RequestDate>2022-01-15</RequestDate>
  <ResponseDueDate>2022-01-30</ResponseDueDate>

  <!--Claim Details-->
  <ClaimantJobTitle>DataAnalyst</ClaimantJobTitle>
  <ClaimantReportedFirstDayofWork>2018-09-
20</ClaimantReportedFirstDayofWork>
  <ClaimantReportedLastDayofWork>2021-09-
20</ClaimantReportedLastDayofWork>
  <ClaimantSepReasonCode>1</ClaimantSepReasonCode>

  <!--Weeks and Wages-->
  <WagesWeeksNeededCode>WO</WagesWeeksNeededCode>
  <WagesNeededBeginDate>2021-01-20</WagesNeededBeginDate>
  <WagesNeededEndDate>2021-08-20</WagesNeededEndDate>

  <!-- Attachment-->
  <AttachmentOccurrence>
    <UniqueAttachmentId>01</UniqueAttachmentId>
    <DescriptionofAttachmentCode>1</DescriptionofAttachmentCode>
    <TypeofDocument>Separation Request</TypeofDocument>
    <ActionableAttachment>1</ActionableAttachment>
    <AttachmentSize>1500</AttachmentSize>
    <AttachmentData>MA==</AttachmentData>
  </AttachmentOccurrence>
</StateSeparationRequest>

```

#### 4.4.1.1.1.2 Central Broker Acknowledgement to State Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of requests it received, the number in error, the dates that it started receiving the records and finished receiving the records, and the BrokerRecordTransactionNumbers. The BrokerRecordTransactionNumbers is a repeating group of State Request File GUIDs and Broker Record Transaction Numbers for each successful record that can then be used by the State in the Additional Fact-Finding request record. This verifies to the State that the Broker received the desired file so it can move on to the next file.

#### REDACTED

```
<StateRequestFileGUID
xmlns="https://uidataexchange.org/schemas">01234567890123456789012345678912</StateRe
questFileGUID>
  <NumberOfRequestRecordsReceived
xmlns="https://uidataexchange.org/schemas">1</NumberOfRequestRecordsReceived>
  <NumberOfRequestRecordsInError
xmlns="https://uidataexchange.org/schemas">0</NumberOfRequestRecordsInError>
  <DateStartedReceivingTransmission xmlns="https://uidataexchange.org/schemas">2022-02-
07T14:11:53.14-05:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission xmlns="https://uidataexchange.org/schemas">2022-02-
07T14:11:53.175-05:00</DateFinishedReceivingTransmission>
  <BrokerRecordTransactionNumbers xmlns="https://uidataexchange.org/schemas">
    <BRTN>
      <StateRequestRecordGUID>1100000000000000000000000000000001
    </StateRequestRecordGUID>
      <BrokerRecordTransactionNumber>105679551</BrokerRecordTransactionNumber>
    </BRTN>
  </BrokerRecordTransactionNumbers>
</StateSeparationRequestCollectionAcknowledgement>
```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the “Post” message, the Broker will also return to the State all of the information that it can on why each individual request failed. The FailedSeparationRequest element defined in the REDACTED XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```
<FailedSeparationRequest>
  <StateRequestRecordGUID>0000000000000000000000000000000099108</StateRequestRecordGUI
D>
  <ErrorOccurrence>
    <ErrorCode>1201</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
```



```
</ErrorOccurrence>
</FailedSeparationRequest>
```

Putting this together with the successful acknowledgement:

**REDACTED**

```
<StateRequestFileGUID>D0F7202142A448F0747E99F75CE0FC00</StateRequestFileGUID>
<FailedSeparationRequest>
  <StateRequestRecordGUID>00000000000000000000000000000099108</StateRequestRecordGUID>
D>
  <ErrorOccurrence>
    <ErrorCode>1201</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedSeparationRequest>
<NumberOfRequestRecordsReceived>2</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>1</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2009-07-13T02:37:53.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2009-07-13T02:37:54.000-
04:00</DateFinishedReceivingTransmission>
<BrokerRecordTransactionNumbers xmlns="https://uidataexchange.org/schemas">
  <BRTN>
    <StateRequestRecordGUID>110000000000000000000000000000001
</StateRequestRecordGUID>
    <BrokerRecordTransactionNumber>105679551</BrokerRecordTransactionNumber>
  </BRTN>
</BrokerRecordTransactionNumbers>
```

#### 4.4.1.1.2 Integrated Partner Post Payload

##### 4.4.1.1.2.1 Post to Central Broker Payload

The "Post" payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED** xsd

**REDACTED**

```
<EmployerTPASeparationResponse>
  <!--Header-->
  <StateRequestRecordGUID>110000000000000000000000000000001</StateRequestRecordGUID>
  <BrokerRecordTransactionNumber>105554582</BrokerRecordTransactionNumber>
  <!-- Employer Information -->
  <StateEmployerAccountNbr>12000120001200012000</StateEmployerAccountNbr>
  <CorrectedStateEmployerAccountNbr>12000120001200013000</CorrectedStateEmployerAccountNbr>
  <EmployerName>TV Studio Enterprises</EmployerName>
```

```

<CorrectedFEIN>123456999</CorrectedFEIN>

<!-- Claimant Information -->
<SSN>123456666</SSN>
<OtherSSN>123456688</OtherSSN>
<ClaimantLastName>Bunny</ClaimantLastName>
<ClaimantFirstName>Bugs</ClaimantFirstName>
<ClaimantMiddleInitial>A</ClaimantMiddleInitial>
<ClaimantSuffix>Jr</ClaimantSuffix>
<ClaimantNameWorkedAsForEmployer>Bugs
Bunny</ClaimantNameWorkedAsForEmployer>

<!-- Claim Data -->
<ClaimEffectiveDate>2021-01-15</ClaimEffectiveDate>
<ClaimNumber>1234</ClaimNumber>
<RequestDate>2022-01-01</RequestDate>
<ResponseDueDate>2022-01-15</ResponseDueDate>

<!-- Claim Details -->
<FraudInd>Y</FraudInd>
<FraudReasons>1</FraudReasons>
<ContractEmployee1099Ind>Y</ContractEmployee1099Ind>
<SeasonalEmploymentInd>Y</SeasonalEmploymentInd>
<EmployerSepReasonCode>1</EmployerSepReasonCode>
<ClaimantJobTitle>Data Analyst</ClaimantJobTitle>
<EmployerReportedClaimantFirstDayofWork>2018-09-
05</EmployerReportedClaimantFirstDayofWork>
<EmployerReportedClaimantLastDayofWork>2021-03-
05</EmployerReportedClaimantLastDayofWork>

<!-- Return to Work Information & Working Available Hours -->
<ReturnToWorkDateInd>Y</ReturnToWorkDateInd>
<ReturnToWorkDate>2021-11-04</ReturnToWorkDate>
<ClaimantReportedReturnToWorkDate>2021-07-
15</ClaimantReportedReturnToWorkDate>

<!-- Weeks and Wages -->
<TotalEarnedWagesNeededInd>1</TotalEarnedWagesNeededInd>
<TotalEarnedWages>120000</TotalEarnedWages>
<TotalWeeksWorkedNeededInd>1</TotalWeeksWorkedNeededInd>
<TotalWeeksWorked>4</TotalWeeksWorked>

<WagesEarnedAfterClaimEffectiveDate>48326</WagesEarnedAfterClaimEffectiveDate>

<NumberOfHoursWorkedAfterClaimEffectiveDate>2320</NumberOfHoursWorkedAfterCl
aimEffectiveDate>

<!-- Remuneration -->
<RemunerationInd>N</RemunerationInd>

<!-- Pension & 401K -->

```

```

<ReceivePension401KInd>4</ReceivePension401KInd>

<!-- Attachment -->
<AdditionalSeparationAckInd>Y</AdditionalSeparationAckInd>
<AttachmentInd>Y</AttachmentInd>
<AttachmentOccurrence>
  <UniqueAttachmentId>01</UniqueAttachmentId>
  <DescriptionofAttachmentCode>2</DescriptionofAttachmentCode>
  <TypeofDocument>Response to Separation Notice</TypeofDocument>
  <AttachmentSize>1000</AttachmentSize>
  <AttachmentData>MA==</AttachmentData>
</AttachmentOccurrence>

<!-- Preparer Information -->
<PreparerTypeCode>E</PreparerTypeCode>
<PreparerTelephoneNumberPlusExt>70311122221234</PreparerTelephoneNumberPlusExt
>

  <PreparerContactName>Debraaa</PreparerContactName>
  <PreparerTitle>HR Manager</PreparerTitle>
  <PreparerEmailAddress>hr@xxabcdef.com</PreparerEmailAddress>
</EmployerTPASeparationResponse>

```

#### 4.4.1.1.2.2 Central Broker Acknowledgement to Integrated Partner Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of responses it received, the number in error, and the dates that it started receiving the records and finished receiving the records. This verifies to the Integrated Partner that the Broker received the desired file so it can move on to the next file.

#### REDACTED

```

<EmployerTPAResponseFileGUID>542A4AE2395FEDDF1EAA3E57F2DFBCE0
</EmployerTPAResponseFileGUID>
<NumberOfResponseRecordsReceived>22</NumberOfResponseRecordsReceived>
<NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
<DateStartedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateFinishedReceivingTransmission>
</EmployerTPASeparationResponseCollectionAcknowledgement>

```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the "Post" message, the Broker will also return to the Integrated Partner all of the information that it can on why each individual response failed. The

FailedSeparationResponse element defined in the **REDACTED** XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```

<FailedSeparationResponse>
  <StateRequestRecordGUID>000000000000000000000000099943
</StateRequestRecordGUID>
  <BrokerRecordTransactionNumber>2001552
</BrokerRecordTransactionNumber>
  <ErrorOccurrence>
    <ErrorCode>2442</ErrorCode>
    <ErrorMessage>Business Rule violation - TotalWeeksWorkedNeededInd (B-62) required if
WagesWeeksNeededCode (A-22) is WW and EmployerSeparationReasonCode (B-20) does not
equal 20 or 21 or 23 or is null.</ErrorMessage>
  </ErrorOccurrence>
</FailedSeparationResponse>

```

Putting this together with the successful acknowledgement:

#### **REDACTED**

```

<EmployerTPAResponseFileGUID>542A4AE2395FEDDF1EAA3E57F2DFBCE0</EmployerTPARespo
nseFileGUID>

<FailedSeparationResponse>
  <StateRequestRecordGUID>000000000000000000000000099943</StateRequestRecordGUI
D>
  <BrokerRecordTransactionNumber>2001552</BrokerRecordTransactionNumber>
  <ErrorOccurrence>
    <ErrorCode>2442</ErrorCode>
    <ErrorMessage>Business Rule violation - TotalWeeksWorkedNeededInd (B-62) required if
WagesWeeksNeededCode (A-22) is WW and EmployerSeparationReasonCode (B-20) does not
equal 20 or 21 or 23 or is null.</ErrorMessage>
  </ErrorOccurrence>
</FailedSeparationResponse>
  <NumberOfResponseRecordsReceived>22</NumberOfResponseRecordsReceived>
  <NumberOfResponseRecordsInError>1</NumberOfResponseRecordsInError>
  <DateStartedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateFinishedReceivingTransmission>
</EmployerTPASeparationResponseCollectionAcknowledgement>

```

#### **4.4.1.2 Pull Payload**

A “Pull” is defined (see Section 4.1.2) as sending a query to the Central Broker to allow the Central Broker to send the Connector all of the requests or responses it has waiting for it (in multiple transactions if they are from different endpoints or if the data exceeds 8MB). This section discusses the (pre-encryption) payloads in the SOAP message.

#### 4.4.1.2.1 State Pull Payload

##### 4.4.1.2.1.1 Pull from Central Broker

The StateSeparationResponseCollectionQuery defined in the **REDACTED** WSDL (see Section 4.6) is a complex type that allows the caller to specify one of three operations: a Pull, a Re-Pull by StateSOAPTransactionNumber, and a Re-Pull by a Date Range.

<!-- Query element for states to collect claim responses they are expecting -->  
**REDACTED**

For the regular Pull, the caller needs to supply only the State Unique ID. Although there are different ways to verify the calling State besides this element, the Broker uses it as an additional security check. Also, there is a requirement in the WSDL 1.1 specification that a WSDL definition have at least one input attribute.

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
</StateSeparationResponseCollectionQuery>
```

For the Re-Pull by StateSOAPTransactionNumber, the caller needs to supply the State Unique ID and the StateSOAPTransactionNumber element out of the StateSeparationResponseCollectionQueryCriteriaType. This will allow the Broker to send the file defined by the StateSOAPTransactionNumber.

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
<StateSeparationResponseCollectionQueryCriteria>  
  <StateSOAPTransactionNumber>123456221</StateSOAPTransactionNumber>  
</StateSeparationResponseCollectionQueryCriteria>  
</StateSeparationResponseCollectionQuery>
```

For the Re-Pull by date range, the caller needs to supply the State Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and a StateSOAPTransactionNumber.

The first time this operation is called, the StateSOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next StateSOAPTransactionNumber that it sent during that date range (in element name NextStateSOAPTransactionNumber).

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

In the next call to this operation, the caller must include the NextStateSOAPTransactionNumber as the StateSOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next StateSOAPTransactionNumber (so there will not be a NextStateSOAPTransactionNumber included in the http response SOAP header).

First Call:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
<StateSeparationResponseCollectionQueryCriteria>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>
</StateSeparationResponseCollectionQueryCriteria>
```

Subsequent Calls:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
<StateSeparationResponseCollectionQueryCriteria>
  <StateSOAPTransactionNumber>12345678901234567890123456789012</StateSOAPTransa
ctionNumber>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-
04:00</BrokerRecordEffectiveDateTo>
</StateSeparationResponseCollectionQueryCriteria>
```

#### 4.4.1.2.1.2 Central Broker Response to the State

When the Broker receives a “Pull” request from a State, it begins assembling all the responses that are intended for that State. It constructs a SOAP message according to the rules for a Separation Information SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds the additional field BrokerRecordEffectiveDate to the response, which indicates the date that the response was accepted into the Broker. It then sends the responses in the HTTP response.

**REDACTED**

```
<StateSeparationResponse>
```

```
<!--Header-->
```

```

<StateRequestRecordGUID>11000000000000000000000000000001</StateRequestRecord
GUID>

<BrokerRecordTransactionNumber>105554582</BrokerRecordTransactionNumber>

<!-- Employer Information -->
<StateEmployerAccountNbr>12000120001200012000</StateEmployerAccountNbr>
<CorrectedStateEmployerAccountNbr>12000120001200013000</CorrectedStateEmploye
rAccountNbr>
  <EmployerName>TV Studio Enterprises</EmployerName>
  <CorrectedFEIN>123456999</CorrectedFEIN>

<!-- Claimant Information -->
<SSN>123456666</SSN>
<OtherSSN>123456688</OtherSSN>
<ClaimantLastName>Bunny</ClaimantLastName>
<ClaimantFirstName>Bugs</ClaimantFirstName>
<ClaimantMiddleInitial>A</ClaimantMiddleInitial>
<ClaimantSuffix>Jr</ClaimantSuffix>
<ClaimantNameWorkedAsForEmployer>Bugs
Bunny</ClaimantNameWorkedAsForEmployer>

<!-- Claim Data -->
<ClaimEffectiveDate>2021-01-15</ClaimEffectiveDate>
<ClaimNumber>1234</ClaimNumber>
<RequestDate>2022-01-01</RequestDate>
<ResponseDueDate>2022-01-15</ResponseDueDate>

<!-- Claim Details -->
<FraudInd>Y</FraudInd>
<FraudReasons>1</FraudReasons>
<ContractEmployee1099Ind>Y</ContractEmployee1099Ind>
<SeasonalEmploymentInd>Y</SeasonalEmploymentInd>
<EmployerSepReasonCode>1</EmployerSepReasonCode>
<ClaimantJobTitle>Data Analyst</ClaimantJobTitle>
<EmployerReportedClaimantFirstDayofWork>2018-09-
05</EmployerReportedClaimantFirstDayofWork>
  <EmployerReportedClaimantLastDayofWork>2021-03-
05</EmployerReportedClaimantLastDayofWork>

<!-- Return to Work Information & Working Available Hours -->
<ReturnToWorkDateInd>Y</ReturnToWorkDateInd>
<ReturnToWorkDate>2021-11-04</ReturnToWorkDate>
<ClaimantReportedReturnToWorkDate>2021-07-
15</ClaimantReportedReturnToWorkDate>

<!-- Weeks and Wages -->
<TotalEarnedWagesNeededInd>1</TotalEarnedWagesNeededInd>
<TotalEarnedWages>120000</TotalEarnedWages>
<TotalWeeksWorkedNeededInd>1</TotalWeeksWorkedNeededInd>
<TotalWeeksWorked>4</TotalWeeksWorked>

```

```

    <WagesEarnedAfterClaimEffectiveDate>48326</WagesEarnedAfterClaimEffectiveDate>

    <NumberOfHoursWorkedAfterClaimEffectiveDate>2320</NumberOfHoursWorkedAfterCl
aimEffectiveDate>

    <!-- Remuneration -->
    <RemunerationInd>N</RemunerationInd>

    <!-- Pension & 401K -->
    <ReceivePension401KInd>4</ReceivePension401KInd>

    <!-- Attachment -->
    <AdditionalSeparationAckInd>Y</AdditionalSeparationAckInd>
    <AttachmentInd>Y</AttachmentInd>
    <AttachmentOccurrence>
        <UniqueAttachmentId>01</UniqueAttachmentId>
        <DescriptionofAttachmentCode>2</DescriptionofAttachmentCode>
        <TypeofDocument>Response to Separation Notice</TypeofDocument>
        <AttachmentSize>1000</AttachmentSize>
        <AttachmentData>MA==</AttachmentData>
    </AttachmentOccurrence>

    <!-- Preparer Information -->
    <PreparerTypeCode>E</PreparerTypeCode>
    <PreparerTelephoneNumberPlusExt>70311122221234</PreparerTelephoneNumberPlusExt
>
    <PreparerContactName>Debraaa</PreparerContactName>
    <PreparerTitle>HR Manager</PreparerTitle>
    <PreparerEmailAddress>hr@xxabcdef.com</PreparerEmailAddress>
    <BrokerRecordEffectiveDate>2009-07-22T01:39:20.000-04:00</BrokerRecordEffectiveDate>
</StateSeparationResponse>

```

#### 4.4.1.2.1.3 State Acknowledgement to the Central Broker

The **REDACTED** is initiated once the State has received its file from the Broker. The acknowledgement must accompany every State Pull request, as this is the manner in which the Broker knows that the State Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next "Pull" call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the State until the NASWA SIDES team and State Administrator can work out the problem.

The key field in this message is the StateSOAPTransmissionNumber, which must correspond with the StateSOAPTransactionNumber sent back in the Broker



Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the State does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

## REDACTED

```
<StateSOAPTransmissionNumber>12345678901234567890123456789012</StateSOAPTransmissionNumber>
  <NumberOfResponseRecordsReceived>0</NumberOfResponseRecordsReceived>
  <NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
  <DateStartedReceivingTransmission>2008-12-31T12:00:00.000-04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2008-12-31T12:00:00.000-04:00</DateFinishedReceivingTransmission>
</StateSeparationResponseCollectionAcknowledgement>
```

### 4.4.1.2.2 Integrated Partner Pull

#### 4.4.1.2.2.1 Pull from Central Broker

The **REDACTED** is a complex query type that allows the caller to specify one of three operations: a Pull, a Re-Pull by EmployerTPASOAPTransactionNumber, and a Re-Pull by a Date Range.

```
<!-- Query element for employer to collect claim responses they are expecting -->
  < REDACTED

  <!-- Types for query element for Employers/TPAs to collect claim requests they are expecting -->
>
```

For the regular Pull, the caller needs to supply only the Integrated Partner's Unique ID. Although there are different ways to verify the calling Integrated Partner besides this element, the Broker uses it as an additional security check. Also, there is a requirement in WSDL 1.1 that a WSDL definition have at least one input attribute.

## REDACTED

```
<UniqueID>BR000000001</UniqueID>
</EmployerTPASeparationRequestCollectionQuery>
```

For the Re-Pull by EmployerTPASOAPTransactionNumber, the caller needs to supply the Integrated Partner Unique ID and the EmployerTPASOAPTransactionNumber element out of the EmployerTPASeparationResponseCollectionQueryCriteriaType. This will allow the Broker to send the file defined by the EmployerTPASOAPTransactionNumber.

## REDACTED

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPASeparationRequestCollectionQueryCriteria>
  <EmployerTPASOAPTransactionNumber>26151</EmployerTPASOAPTransactionNumber>
</EmployerTPASeparationRequestCollectionQueryCriteria>
</EmployerTPASeparationRequestCollectionQuery>
```

For the Re-Pull by date range, the caller needs to supply the Integrated Partner Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex query type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and an EmployerTPASOAPTransactionNumber.

The first time this operation is called, the EmployerTPASOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next EmployerTPASOAPTransactionNumber that it sent during that date range (in element name NextEmployerTPASOAPTransactionNumber). In the next call to this operation, the caller must include the NextEmployerTPASOAPTransactionNumber as the EmployerTPASOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next EmployerTPASOAPTransactionNumber (so there will not be a NextEmployerTPASOAPTransactionNumber included in the http response SOAP header).

First Call:

### **REDACTED**

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPASeparationRequestCollectionQueryCriteria>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>
</EmployerTPASeparationRequestCollectionQueryCriteria>
```

Subsequent Calls:

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>  
<EmployerTPASeparationRequestCollectionQueryCriteria>  
  <EmployerTPASOAPTransactionNumber>12345678901234567890123456789012</Employer  
TPASOAPTransactionNumber>  
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-  
04:00</BrokerRecordEffectiveDateFrom>  
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-  
04:00</BrokerRecordEffectiveDateTo>  
</EmployerTPASeparationRequestCollectionQueryCriteria>
```

**4.4.1.2.2.2 Central Broker Response to Integrated Partner**

When the Broker receives a “Pull” request from an Integrated Partner, it begins assembling all the requests that are intended for that Integrated Partner. It constructs a SOAP message according to the rules for a Separation Information SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds two additional fields to the Separation Request - the BrokerRecordEffectiveDate and the BrokerRecordTransactionNumber. The BrokerRecordEffectiveDate indicates the date that the request was accepted into the Broker. The BrokerRecordTransactionNumber is a unique record tracking number and must be returned on the response for this record. It then sends the separation requests in the HTTP response.

**REDACTED**

```
<EmployerTPASeparationRequest>  
  <StateRequestRecordGUID>11000000000000000000000000000001</StateRequestRecordG  
UID>  
  <!--State Info-->  
  <RequestingStateAbbreviation>VA</RequestingStateAbbreviation>  
  <UIOfficeName>BU1</UIOfficeName>  
  <UIOfficePhone>5553247787</UIOfficePhone>  
  <UIOfficeFax>5553247787</UIOfficeFax>  
  
  <UIOfficeEmailAddress>customerservice@vec.virginia.gov</UIOfficeEmailAddress>  
  <AllocationRequiredInd>Y</AllocationRequiredInd>  
  
  <!--Employer Information-->  
  <StateEmployerAccountNbr>12000120001200012000</StateEmployerAccountNbr>  
  <EmployerName>FedPoint Systems LLC</EmployerName>  
  <FEIN>123456789</FEIN>  
  <TypeofEmployerCode>3</TypeofEmployerCode>  
  
  <!--Claimant Information-->  
  <SSN>123456666</SSN>  
  <ClaimantLastName>WWisWO_AllocIndisY</ClaimantLastName>  
  <OtherLastName>Bugs</OtherLastName>  
  <ClaimantFirstName>AttachIndls1</ClaimantFirstName>
```

```

<ClaimantSuffix>Jr</ClaimantSuffix>

<!--Claim Data-->
<ClaimEffectiveDate>2021-07-30</ClaimEffectiveDate>
<TypeofClaimCode>3</TypeofClaimCode>
<BenefitYearBeginDate>2021-06-04</BenefitYearBeginDate>
<RequestDate>2022-01-15</RequestDate>
<ResponseDueDate>2022-01-30</ResponseDueDate>

<!--Claim Details-->
<ClaimantJobTitle>DataAnalyst</ClaimantJobTitle>
<ClaimantReportedFirstDayofWork>2018-09-
20</ClaimantReportedFirstDayofWork>
<ClaimantReportedLastDayofWork>2021-09-
20</ClaimantReportedLastDayofWork>
<ClaimantSepReasonCode>1</ClaimantSepReasonCode>

<!--Weeks and Wages-->
<WagesWeeksNeededCode>WO</WagesWeeksNeededCode>
<WagesNeededBeginDate>2021-01-20</WagesNeededBeginDate>
<WagesNeededEndDate>2021-08-20</WagesNeededEndDate>

<!-- Attachment-->
<AttachmentOccurrence>
  <UniqueAttachmentId>01</UniqueAttachmentId>
  <DescriptionofAttachmentCode>1</DescriptionofAttachmentCode>
  <TypeofDocument>Separation Request</TypeofDocument>
  <ActionableAttachment>1</ActionableAttachment>
  <AttachmentSize>1500</AttachmentSize>
  <AttachmentData>MA==</AttachmentData>
</AttachmentOccurrence>
<BrokerRecordTransactionNumber>2001569</BrokerRecordTransactionNumber>
<BrokerRecordEffectiveDate>2009-07-13T14:35:58.000-04:00</BrokerRecordEffectiveDate>
</EmployerTPASeparationRequest>

```

#### 4.4.1.2.2.3 Integrated Partner Acknowledgement to Central Broker

The **REDACTED** is initiated once the Integrated Partner has received its file from the Broker. The acknowledgement must accompany every Integrated Partner Pull request, as this is the manner in which the Broker knows that the Integrated Partner Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next "Pull" call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the Integrated Partner until the NASWA SIDES team and Integrated Partner Administrator can work out the problem.

The key field in this message is the EmployerTPASOAPTransmissionNumber, which must correspond with the EmployerTPASOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the Integrated Partner does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

### REDACTED

```
<EmployerTPASOAPTransmissionNumber>3211</EmployerTPASOAPTransmissionNumber>
<NumberOfRequestRecordsReceived>4</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateFinishedReceivingTransmission>
</EmployerTPASeparationRequestCollectionAcknowledgement>
```

## 4.4.2 Earnings Verification

### 4.4.2.1 Post Payload

A “Post” is defined (see Section 4.1.1) as sending a request or response to the Broker by a particular Connector. This section discusses the (pre-encryption) payloads in the SOAP message for the Earnings Verification Exchange.

#### 4.4.2.1.1 State Post Payload

##### 4.4.2.1.1.1 Post to Central Broker Payload

The “Post” payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED** xsd.

### REDACTED

```
<StateEarningsVerificationRequest>
  <StateEarningsVerificationRequestRecordGUID>AAA53000000000000000000000000000003</S
tateEarningsVerificationRequestRecordGUID>
  <RequestingStateAbbreviation>ST</RequestingStateAbbreviation>
  <UIOfficeName>Office Name</UIOfficeName>
  <UIOfficePhone>5555555555</UIOfficePhone>
  <UIOfficeFax>5555555554</UIOfficeFax>
  <UIOfficeEmailAddress>james.madison@state.gov</UIOfficeEmailAddress>
  <StateEmployerAccountNbr>1234567890</StateEmployerAccountNbr>
  <FEIN>123456789</FEIN>
  <EmployerName>ACME</EmployerName>
  <SSN>311111334</SSN>
  <ClaimantLastName>Lastname</ClaimantLastName>
  <ClaimantFirstName>Firstname</ClaimantFirstName>
  <ClaimantMiddleInitial>M</ClaimantMiddleInitial>
  <ClaimantSuffix>JR</ClaimantSuffix>
  <EarningsVerificationWeekBeginDate>2010-08-01</EarningsVerificationWeekBeginDate>
  <EarningsVerificationWeekEndDate>2010-08-07</EarningsVerificationWeekEndDate>
```

```

    <EarningsVerificationComments>This is a comment field for this Earnings Verification
Request</EarningsVerificationComments>
    <RequestDate>2010-10-14</RequestDate>
    <EarningsStatusCode>3</EarningsStatusCode>
    <TipsStatusCode>2</TipsStatusCode>
    <CommissionStatusCode>2</CommissionStatusCode>
    <BonusStatusCode>2</BonusStatusCode>
    <VacationStatusCode>2</VacationStatusCode>
    <SickLeaveStatusCode>2</SickLeaveStatusCode>
    <HolidayStatusCode>3</HolidayStatusCode>
    <SeveranceStatusCode>3</SeveranceStatusCode>
    <WagesInLieuStatusCode>4</WagesInLieuStatusCode>
    <EarningsVerificationResponseCommentIndicator>1</EarningsVerificationResponseCom
mentIndicator>
    <ResponseDueDate>2010-10-28</ResponseDueDate>
    <EarningsVerificationSourceCode>9</EarningsVerificationSourceCode>
</StateEarningsVerificationRequest>

```

#### 4.4.2.1.1.2 Central Broker Acknowledgement to State Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of requests it received, the number in error, and the dates that it started receiving the records and finished receiving the records. This verifies to the State that the Broker received the desired file so it can move on to the next file.

#### < REDACTED

```

    <StateRequestFileGUID>C0EB9C5D24CA4B8EFB8AEED97A5252C8</StateRequestFileGUID>
    <NumberOfRequestRecordsReceived>1</NumberOfRequestRecordsReceived>
    <NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
    <DateStartedReceivingTransmission>2010-11-18T10:26:52.730-
05:00</DateStartedReceivingTransmission>
    <DateFinishedReceivingTransmission>2010-11-18T10:26:54.610-
05:00</DateFinishedReceivingTransmission>

```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the “Post” message, the Broker will also return to the State all of the information that it can on why each individual request failed. The FailedEarningVerificationRequest element defined in the **REDACTED** request XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```

    <FailedEarningsVerificationRequest>
    <StateEarningsVerificationRequestRecordGUID>31101000000000000000000000000001</State
EarningsVerificationRequestRecordGUID>
    <ErrorOccurrence>

```

```

    <ErrorCode>311</ErrorCode>
    <ErrorMessage>Business Rule violation - The NumberofWeeksRequested (Character - Size
2) must equal the number of days included in EarningsVerificationWeekBeginDate thru
EarningsVerificationWeekEndDate divided by 7 days.</ErrorMessage>
  </ErrorOccurrence>
</FailedEarningsVerificationRequest>

```

Putting this together with the successful acknowledgement:

**REDACTED**

```

<StateRequestFileGUID>9CEBA17833DFD7C6476EEDE25D20FFF6</StateRequestFileGUID>
<FailedEarningsVerificationRequest>
  <StateEarningsVerificationRequestRecordGUID>311010000000000000000000000001</State
EarningsVerificationRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>311</ErrorCode>
    <ErrorMessage>Business Rule violation - The NumberofWeeksRequested (Character - Size
2) must equal the number of days included in EarningsVerificationWeekBeginDate thru
EarningsVerificationWeekEndDate divided by 7 days.</ErrorMessage>
  </ErrorOccurrence>
</FailedEarningsVerificationRequest>
<NumberOfRequestRecordsReceived>1</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>1</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2011-01-21T10:49:27.929-
05:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2011-01-21T10:49:28.181-
05:00</DateFinishedReceivingTransmission>
</StateEarningsVerificationRequestCollectionAcknowledgement>

```

**4.4.2.1.2 Integrated Partner Post Payload**

**4.4.2.1.2.1 Post to Central Broker Payload**

The “Post” payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED** xsd.

**REDACTED**

```

<EmployerTPAEarningsVerificationResponse>
  <!-- Backfilled -->

  <StateEarningsVerificationRequestRecordGUID>AAA53000000000000000000000000003</S
tateEarningsVerificationRequestRecordGUID>
  <!-- Backfilled -->
  <BrokerRecordTransactionNumber>5447</BrokerRecordTransactionNumber>
  <!-- Backfilled -->
  <RequestDate>2010-09-01</RequestDate>
  <!-- Backfilled -->
  <ResponseDueDate>2010-09-15</ResponseDueDate>
  <!-- Backfilled -->
  <RequestingStateAbbreviation>ST</RequestingStateAbbreviation>

```

```

<!-- Backfilled -->
<UIOfficeName>Office Name</UIOfficeName>
<!-- Backfilled -->
<StateEmployerAccountNbr>1234567890</StateEmployerAccountNbr>
<!-- Backfilled -->
<FEIN>123456789</FEIN>
<CorrectedFEIN>987654321</CorrectedFEIN>
<!-- Backfilled -->
<EmployerName>ACME</EmployerName>
<CorrectedEmployerName>Fly By Night</CorrectedEmployerName>
<!-- Backfilled -->
<SSN>311111334</SSN>
<!-- Backfilled -->
<ClaimantLastName>Lastname</ClaimantLastName>
<!-- Backfilled -->
<ClaimantFirstName>Firstname</ClaimantFirstName>
<!-- Backfilled -->
<ClaimantMiddleInitial>M</ClaimantMiddleInitial>
<!-- Backfilled -->
<ClaimantSuffix>JR</ClaimantSuffix>
<ClaimantNameWorkedAsForEmployer>John Q
Public</ClaimantNameWorkedAsForEmployer>
<!-- Backfilled -->
<EarningsVerificationWeekBeginDate>2010-08-01</EarningsVerificationWeekBeginDate>
<!-- Backfilled -->
<EarningsVerificationWeekEndDate>2010-08-07</EarningsVerificationWeekEndDate>
<!-- 1 - Claimant works, 20 - Never Employed Here, 21 - TPA does not represent Employer -
->
<ClaimantEmployerWorkRelationshipCode>1</ClaimantEmployerWorkRelationshipCode
>
<!-- 1 Yes, has earnings, 2 - did not have earnings (100% Sales), 9 - No Work -->
<EmployerEarningsCode>1</EmployerEarningsCode>
<FirstDayWorkedinPeriod>2010-08-01</FirstDayWorkedinPeriod>
<!-- 1 for Yes, 2 for No -->
<StillWorkingCode>2</StillWorkingCode>
<LastDayWorked>2010-08-02</LastDayWorked>
<!-- 1 - Layoff, 2 - Fired, 3 - Vol Quit, 4 - Other -->
<EmployerSepReasonCode>1</EmployerSepReasonCode>
<!-- When Request = 1 or (2 with Work/Relationship = 20/21 or Earnings Code = 9) -->
<EarningsVerificationResponseComment>This employee was let go during the time
period</EarningsVerificationResponseComment>
<DailyEarningsVerification>
  <EVDAY>2010-08-01</EVDAY>
  <HoursWorked>15:00</HoursWorked>
  <!-- See Request values for required or not for all below -->
  <AmountEarnedForTheDay>500.00</AmountEarnedForTheDay>
  <EarningsPaidDate>2010-08-02</EarningsPaidDate>
</DailyEarningsVerification>
<DailyEarningsVerification>
  <WeekBeginDate>2010-08-08</WeekBeginDate>
  <WeekEndDate>2010-08-14</WeekEndDate>
  <HoursWorked>15:00</HoursWorked>

```



```

    <!-- See Request values for required or not for all below -->
    <AmountEarnedForWeek>500.00</AmountEarnedForWeek>
    <EarningsPaidDate>2010-08-02</EarningsPaidDate>
  </DailyEarningsVerification>

  <!-- E - Employer, T - TPA -->
  <PreparerTypeCode>T</PreparerTypeCode>
  <PreparerCompanyName>ABC TPA</PreparerCompanyName>
  <PreparerTelephoneNumberPlusExt>5555555556</PreparerTelephoneNumberPlusExt>
  <PreparerContactName>Mrs Sue Herman</PreparerContactName>
  <PreparerTitle>Claims Administrator</PreparerTitle>
  <PreparerFaxNbr>5555555557</PreparerFaxNbr>
  <PreparerEmailAddress>sue.herman@abctpa.com</PreparerEmailAddress>
  <!-- Backfilled -->
  <EarningsVerificationSourceCode>9</EarningsVerificationSourceCode>
</EmployerTPAEarningsVerificationResponse>

```

#### 4.4.2.1.2.2 Central Broker Acknowledgement to Integrated Partner Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of responses it received, the number in error, and the dates that it started receiving the records and finished receiving the records. This verifies to the Integrated Partner that the Broker received the desired file so it can move on to the next file.

#### REDACTED

```

  <EmployerTPAResponseFileGUID>399D60FDB970C10C3619DC0B378ABF77</EmployerTPAResponseFileGUID>
  <NumberOfResponseRecordsReceived>1</NumberOfResponseRecordsReceived>
  <NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
  <DateStartedReceivingTransmission>2011-01-21T11:39:45.842-05:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2011-01-21T11:39:46.310-05:00</DateFinishedReceivingTransmission>

```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the "Post" message, the Broker will also return to the Integrated Partner all of the information that it can on why each individual response failed. The **REDACTED** element defined in the earnings verification response XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```

  <FailedEarningsVerificationResponse>
    <StateEarningsVerificationRequestRecordGUID>45200000000000000000000000000001</State

```

```

EarningsVerificationRequestRecordGUID>
  <BrokerRecordTransactionNumber>6591</BrokerRecordTransactionNumber>
  <ErrorOccurrence>
    <ErrorCode>452</ErrorCode>
    <ErrorMessage>Business Rule violation - There must be a value (Numeric - 7.2) for
SeveranceAmountPaidForWeek in Repeatable Weekly Earnings Verification 1 if
SeveranceStatusCode (from Request) = 2 for Field Required, Date Not Required or 3 for Field
Required, Date Required, Date Paid or 4 for Field Required, Date Required, Date
Allocated</ErrorMessage>
  </ErrorOccurrence>
</FailedEarningsVerificationResponse>

```

Putting this together with the successful acknowledgement:

### REDACTED

```

<EmployerTPAResponseFileGUID>CD685018CC4E55949E425F86B83E4687
</EmployerTPAResponseFileGUID>
<FailedEarningsVerificationResponse>
  <StateEarningsVerificationRequestRecordGUID>45200000000000000000000000000001</State
EarningsVerificationRequestRecordGUID>
  <BrokerRecordTransactionNumber>6591</BrokerRecordTransactionNumber>
  <ErrorOccurrence>
    <ErrorCode>452</ErrorCode>
    <ErrorMessage>Business Rule violation - There must be a value (Numeric - 7.2) for
SeveranceAmountPaidForWeek in Repeatable Weekly Earnings Verification 1 if
SeveranceStatusCode (from Request) = 2 for Field Required, Date Not Required or 3 for Field
Required, Date Required, Date Paid or 4 for Field Required, Date Required, Date
Allocated</ErrorMessage>
  </ErrorOccurrence>
</FailedEarningsVerificationResponse>
<NumberOfResponseRecordsReceived>1</NumberOfResponseRecordsReceived>
<NumberOfResponseRecordsInError>1</NumberOfResponseRecordsInError>
<DateStartedReceivingTransmission>2010-12-14T08:18:59.076-
05:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2010-12-14T08:19:07.776-
05:00</DateFinishedReceivingTransmission>

```

#### 4.4.2.2 Pull Payload

A “Pull” is defined (see Section 4.1.2) as sending a query to the Central Broker to allow the Central Broker to send the Connector all of the requests or responses it has waiting for it (in multiple transactions if they are from different endpoints or if the data exceeds 8MB). This section discusses the (pre-encryption) payloads in the SOAP message.

##### 4.4.2.2.1 State Pull Payload

###### 4.4.2.2.1.1 Pull from Central Broker

The StateEarningsVerificationResponseCollectionQuery defined in the State Earnings Verification WSDL (see Section 4.6) is a complex type that allows the

caller to specify one of three operations: a Pull, a Re-Pull by StateSOAPTransactionNumber, and a Re-Pull by a Date Range.

**REDACTED**

For the regular Pull, the caller needs to supply only the State Unique ID. Although there are different ways to verify the calling State besides this element, the Broker uses it as an additional security check. Also, there is a requirement in the WSDL 1.1 specification that a WSDL definition have at least one input attribute.

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
```

For the Re-Pull by StateSOAPTransactionNumber, the caller needs to supply the State Unique ID and the StateSOAPTransactionNumber element out of the **REDACTED** for **REDACTED**. This will allow the Broker to send the file defined by the StateSOAPTransactionNumber.

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
<StateEarningVerificationResponseCollectionQueryCriteria>  
  <StateSOAPTransactionNumber>42153</StateSOAPTransactionNumber>  
</StateEarningsVerificationResponseCollectionQueryCriteria>
```

For the Re-Pull by date range, the caller needs to supply the State Unique ID and the **REDACTED** element out of the **REDACTED** for Earnings Verification. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and a StateSOAPTransactionNumber.

The first time this operation is called, the StateSOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next StateSOAPTransactionNumber that it sent during that date range (in element name NextStateSOAPTransactionNumber). In the next call to this operation, the caller must include the NextStateSOAPTransactionNumber as the StateSOAPTransactionNumber along with the

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions **including** all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next StateSOAPTransactionNumber (so there will not be a NextStateSOAPTransactionNumber included in the http response SOAP header).

First Call:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
<StateEarningsVerificationResponseCollectionQueryCriteria>
  <BrokerRecordEffectiveDateFrom>2009-01-01T00:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T00:00:00.000-04:00</BrokerRecordEffectiveDateTo>
```

Subsequent Calls:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
<StateEarningsVerificationResponseCollectionQueryCriteria>
<StateSOAPTransactionNumber>12345678901234567890123456789012<StateSOAP
TransactionNumber>
  <BrokerRecordEffectiveDateFrom>2009-01-01T00:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T00:00:00.000-04:00</BrokerRecordEffectiveDateTo>
```

#### 4.4.2.2.1.2 Central Broker Response to the State

When the Broker receives a “Pull” request from a State, it begins assembling all the responses that are intended for that State. It constructs a SOAP message according to the rules for a SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds the additional field BrokerRecordEffectiveDate to the response, which indicates the date that the response was accepted into the Broker. It then sends the responses in the HTTP response.

**REDACTED**

```
<StateEarningsVerificationResponse>
  <StateEarningsVerificationRequestRecordGUID>AAA53000000000000000000000000003</S
tateEarningsVerificationRequestRecordGUID>
  <BrokerRecordTransactionNumber>5447</BrokerRecordTransactionNumber>
  <RequestDate>2010-09-01</RequestDate>
  <ResponseDueDate>2010-09-15</ResponseDueDate>
  <RequestingStateAbbreviation>ST</RequestingStateAbbreviation>
```

```

    <UIOfficeName>Office Name</UIOfficeName>
    <StateEmployerAccountNbr>1234567890</StateEmployerAccountNbr>
    <FEIN>123456789</FEIN>
    <CorrectedFEIN>987654321</CorrectedFEIN>
    <EmployerName>ACME</EmployerName>
    <CorrectedEmployerName>Fly By Night</CorrectedEmployerName>
    <SSN>311111334</SSN>
    <ClaimantLastName>Lastname</ClaimantLastName>
    <ClaimantFirstName>Firstname</ClaimantFirstName>
    <ClaimantMiddleInitial>M</ClaimantMiddleInitial>
    <ClaimantSuffix>JR</ClaimantSuffix>
    <ClaimantNameWorkedAsForEmployer>John Q
Public</ClaimantNameWorkedAsForEmployer>
    <EarningsVerificationWeekBeginDate>2010-08-01</EarningsVerificationWeekBeginDate>
    <EarningsVerificationWeekEndDate>2010-08-07</EarningsVerificationWeekEndDate>
    <ClaimantEmployerWorkRelationshipCode>1</ClaimantEmployerWorkRelationshipCode
>
    <EmployerEarningsCode>1</EmployerEarningsCode>
    <FirstDayWorkedinPeriod>2010-08-01</FirstDayWorkedinPeriod>
    <StillWorkingCode>2</StillWorkingCode>
    <LastDayWorked>2010-08-02</LastDayWorked>
    <EmployerSepReasonCode>1</EmployerSepReasonCode>
    <EarningsVerificationResponseComment>This employee was let go during the time
period</EarningsVerificationResponseComment>
    <DailyEarningsVerification>
        <EVDay>2010-08-01</EVDay>
        <HoursWorked>15:00</HoursWorked>
        <AmountEarnedForTheDay>500.00</AmountEarnedForTheDay>
        <EarningsPaidDate>2010-08-02</EarningsPaidDate>
    </DailyEarningsVerification>
    <DailyEarningsVerification>
        <WeekBeginDate>2010-08-08</WeekBeginDate>
        <WeekEndDate>2010-08-14</WeekEndDate>
        <HoursWorked>15:00</HoursWorked>
        <AmountEarnedForWeek>500.00</AmountEarnedForWeek>
        <EarningsPaidDate>2010-08-02</EarningsPaidDate>
    </DailyEarningsVerification>
    <PreparerTypeCode>T</PreparerTypeCode>
    <PreparerCompanyName>ABC TPA</PreparerCompanyName>
    <PreparerTelephoneNumberPlusExt>5555555556</PreparerTelephoneNumberPlusExt>
    <PreparerContactName>Mrs Sue Herman</PreparerContactName>
    <PreparerTitle>Claims Administrator</PreparerTitle>
    <PreparerFaxNbr>5555555557</PreparerFaxNbr>
    <PreparerEmailAddress>sue.herman@abctpa.com</PreparerEmailAddress>
    <EarningsVerificationSourceCode>9</EarningsVerificationSourceCode>
    <BrokerRecordEffectiveDate>2011-02-10T13:01:19.000-
05:00</BrokerRecordEffectiveDate>
    </StateEarningsVerificationResponse>

```

#### 4.4.2.2.1.3 State Acknowledgement to the Central Broker

The **REDACTED** is initiated once the State has received its file from the Broker. The acknowledgement must accompany every State Pull request, as this is the manner in which the Broker knows that the State Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next "Pull" call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the State until the NASWA SIDES team and State Administrator can work out the problem.

The key field in this message is the StateSOAPTransmissionNumber, which must correspond with the StateSOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the State does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

#### **REDACTED**

```
<StateSOAPTransmissionNumber>12345678901234567890123456789012</StateSOAPTransmissionN  
umber>  
  <NumberOfResponseRecordsReceived>0</NumberOfResponseRecordsReceived>  
  <NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>  
  <DateStartedReceivingTransmission>2001-12-31T12:00:00.00-  
04:00</DateStartedReceivingTransmission>  
  <DateFinishedReceivingTransmission>2001-12-31T12:00:00.00-  
04:00</DateFinishedReceivingTransmission>
```

#### 4.4.2.2.2 Integrated Partner Pull

##### 4.4.2.2.2.1 Pull from Central Broker

The **REDACTED** is a complex query types that allow the caller to specify one of three operations: a Pull, a Re-Pull by EmployerTPASOAPTransactionNumber, and a Re-Pull by a Date Range.

#### **REDACTED**

For the regular Pull, the caller needs to supply only the Integrated Partner Unique ID. Although there are different ways to verify the calling Integrated Partner besides this element, the Broker uses it as an additional security check. Also, there is a requirement in WSDL 1.1 that a WSDL definition have at least one input attribute.

#### **REDACTED**

<UniqueID>BR999999999</UniqueID>

For the Re-Pull by EmployerTPASOAPTransactionNumber, the caller needs to supply the Integrated Partner Unique ID and the EmployerTPASOAPTransactionNumber element out of the **REDACTED**. This will allow the Broker to send the file defined by the EmployerTPASOAPTransactionNumber.

### **REDACTED**

```
<UniqueID>BR999999999</UniqueID>  
<EmployerTPAEarningsVerificationRequestCollectionQueryCriteria>  
  <EmployerTPASOAPTransactionNumber>45459</EmployerTPASOAPTransactionNumber>  
</EmployerTPAEarningsVerificationRequestCollectionQueryCriteria>
```

For the Re-Pull by date range, the caller needs to supply the Integrated Partner Unique ID and the **REDACTED** element out of the **REDACTED**. The re-pull by date range will pull all the files that were previously pulled by the Connector during the date range specified.

The **REDACTED** is a complex query type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and an EmployerTPASOAPTransactionNumber.

The first time this operation is called, the EmployerTPASOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next EmployerTPASOAPTransactionNumber that it sent during that date range (in element name NextEmployerTPASOAPTransactionNumber). In the next call to this operation, the caller must include the NextEmployerTPASOAPTransactionNumber as the EmployerTPASOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next EmployerTPASOAPTransactionNumber (so there will not be a NextEmployerTPASOAPTransactionNumber included in the http response SOAP header).

First Call:

```
<?xml version="1.0" encoding="UTF-8"?>
< REDACTED
  <UniqueID>BR999999999</UniqueID>
  <EmployerTPAEarningsVerificationRequestCollectionQueryCriteria>
    <BrokerRecordEffectiveDateFrom>2009-01-01T00:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
    <BrokerRecordEffectiveDateTo>2010-12-31T00:00:00.000-04:00</BrokerRecordEffectiveDateTo>
  </EmployerTPAEarningsVerificationRequestCollectionQueryCriteria>
```

Subsequent Calls:

```
REDACTED
  <UniqueID>BR999999999</UniqueID>
  <EmployerTPAEarningsVerificationRequestCollectionQueryCriteria>
    <EmployerTPASOAPTransactionNumber>26101</EmployerTPASOAPTransactionNumber>
    <BrokerRecordEffectiveDateFrom>2009-01-01T00:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
    <BrokerRecordEffectiveDateTo>2010-12-31T00:00:00.000-04:00</BrokerRecordEffectiveDateTo>
  </EmployerTPAEarningsVerificationRequestCollectionQueryCriteria>
```

#### 4.4.2.2.2 Central Broker Response to Integrated Partner

When the Broker receives a “Pull” request from an Integrated Partner, it begins assembling all the earnings verification requests that are intended for that Integrated Partner. It constructs a SOAP message according to the rules for a SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds two additional fields to the Request - the BrokerRecordEffectiveDate and the BrokerRecordTransactionNumber. The BrokerRecordEffectiveDate indicates the date that the request was accepted into the Broker. The BrokerRecordTransactionNumber is a unique record tracking number and must be returned on the response for this record. It then sends the requests in the HTTP response.

```
REDACTED
  <EmployerTPAEarningsVerificationRequest>
    <StateEarningsVerificationRequestRecordGUID>20150000000000000000000000000001</State
EarningsVerificationRequestRecordGUID>
    <RequestingStateAbbreviation>ST</RequestingStateAbbreviation>
    <UIOfficeName>Office Name</UIOfficeName>
    <UIOfficePhone>5555555555</UIOfficePhone>
    <UIOfficeFax>5555555554</UIOfficeFax>
    <UIOfficeEmailAddress>james.madison@state.gov</UIOfficeEmailAddress>
    <StateEmployerAccountNbr>1234567890</StateEmployerAccountNbr>
    <FEIN>123456789</FEIN>
    <EmployerName>ACME</EmployerName>
```



```

<SSN>213456721</SSN>
<ClaimantLastName>Lastname</ClaimantLastName>
<ClaimantFirstName>Firstname</ClaimantFirstName>
<ClaimantMiddleInitial>M</ClaimantMiddleInitial>
<ClaimantSuffix>JR</ClaimantSuffix>
<EarningsVerificationWeekBeginDate>2010-08-01</EarningsVerificationWeekBeginDate>
<EarningsVerificationWeekEndDate>2010-09-04</EarningsVerificationWeekEndDate>
<EarningsVerificationComments>This is a comment field for this Earnings Verification
Request</EarningsVerificationComments>
<RequestDate>2010-10-14</RequestDate>
<EarningsStatusCode>3</EarningsStatusCode>
<TipsStatusCode>3</TipsStatusCode>
<CommissionStatusCode>3</CommissionStatusCode>
<BonusStatusCode>3</BonusStatusCode>
<VacationStatusCode>3</VacationStatusCode>
<SickLeaveStatusCode>3</SickLeaveStatusCode>
<HolidayStatusCode>3</HolidayStatusCode>
<SeveranceStatusCode>3</SeveranceStatusCode>
<WagesInLieuStatusCode>3</WagesInLieuStatusCode>
<EarningsVerificationResponseCommentIndicator>1</EarningsVerificationResponseComme
ntIndicator>
<ResponseDueDate>2010-10-28</ResponseDueDate>
<EarningsVerificationSourceCode>9</EarningsVerificationSourceCode>
<BrokerRecordTransactionNumber>6609</BrokerRecordTransactionNumber>
<BrokerRecordEffectiveDate>2011-03-11T13:09:20.000-05:00</BrokerRecordEffectiveDate>
</EmployerTPAEarningsVerificationRequest>

```

### Integrated Partner Acknowledgement to Central Broker

The **REDACTED** is initiated once the Integrated Partner has received its file from the Broker. The acknowledgement must accompany every Integrated Partner Pull request, as this is the manner in which the Broker knows that the Integrated Partner Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next "Pull" call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the Integrated Partner until the NASWA SIDES team and Integrated Partner Administrator can work out the problem.

The key field in this message is the EmployerTPASOAPTransmissionNumber, which must correspond with the EmployerTPASOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the Integrated Partner does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

### REDACTED

```

<EmployerTPASOAPTransmissionNumber>3400</EmployerTPASOAPTransmissionNumber>

```

```
<NumberOfRequestRecordsReceived>4</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateFinishedReceivingTransmission>
```

### 4.4.3 Monetary & Potential Charges

#### 4.4.3.1 Post Payload

A "Post" is defined (see Section 4.1.1) as sending a request or response to the Broker by a particular Connector. This section discusses the (pre-encryption) payloads in the SOAP message.

##### 4.4.3.1.1 State Post Payload

###### 4.4.3.1.1.1 Post to Central Broker Payload

The "Post" payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED** xsd.

**REDACTED**

```
<StateMonetaryChargeRequest>
  <StateMonetaryChargeRequestRecordGUID>e755d30ed7dd4662bc0452e9050c0001
</StateMonetaryChargeRequestRecordGUID>
  <ClaimNumber>123456789012345678</ClaimNumber>
  <FormNumber>form12345678</FormNumber>
  <RequestDate>2019-04-10</RequestDate>
  <ResponseDueDate>2019-05-05</ResponseDueDate>
  <RequestingStateAbbreviation>CA</RequestingStateAbbreviation>
  <UIOfficeName>CA UIOfficeName</UIOfficeName>
  <UIOfficePhone>2101234567</UIOfficePhone>
  <UIOfficeFax>2101234568</UIOfficeFax>
  <UIOfficeEmailAddress>UIOfficeEmailAddress@asdf.gov</UIOfficeEmailAddress>
  <FEIN>999999999</FEIN>
  <StateEmployerAccountNbr>0123456789</StateEmployerAccountNbr>
  <PredecessorEmployerAccountNbr>0123456789</PredecessorEmployerAccountNbr>
  <EmployerName>Test Employer</EmployerName>
  <EmployerAccountType>1</EmployerAccountType>
  <TypeofEmployerCode>1</TypeofEmployerCode>
  <SSN>123000000</SSN>
  <ClaimantLastName>Doe</ClaimantLastName>
  <OtherLastName>Mouse</OtherLastName>
  <ClaimantFirstName>Jane</ClaimantFirstName>
  <ClaimantMiddleInitial>A</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>
  <Dependents>2</Dependents>
  <ClaimantSepReasonCode>2</ClaimantSepReasonCode>
  <BenefitYearBeginDate>2013-07-23</BenefitYearBeginDate>
  <BasePeriodStartDate>2010-07-24</BasePeriodStartDate>
```

```

<BasePeriodEndDate>2011-07-23</BasePeriodEndDate>
<NumberEmployersInBasePeriod>1</NumberEmployersInBasePeriod>
<BasePeriodWagesAllEmployers>42000.00</BasePeriodWagesAllEmployers>
<BasePeriodWagesThisEmployer>36000.00</BasePeriodWagesThisEmployer>
<WeeklyBenefitAmount>1000.00</WeeklyBenefitAmount>
<MaximumBenefitAmount>60000.00</MaximumBenefitAmount>
<TotalWeeksOfBenefits>52</TotalWeeksOfBenefits>
<PercentChargableEmployer>100.00</PercentChargableEmployer>
<TotalEmployerCharge>50000.00</TotalEmployerCharge>
  <EmployerChargeOccurrence>
    <BeginWeekNumber>1</BeginWeekNumber>
    <EndWeekNumber>1</EndWeekNumber>
    <TotalWeeklyEmployerCharge>1000.00</TotalWeeklyEmployerCharge>
    <PotentialWeeksEmployerCharge>10000.00</PotentialWeeksEmployerCharge>
  </EmployerChargeOccurrence>
  <EmployerChargeOccurrence>
    <BeginWeekNumber>2</BeginWeekNumber>
    <EndWeekNumber>2</EndWeekNumber>
    <TotalWeeklyEmployerCharge>1200.00</TotalWeeklyEmployerCharge>
    <PotentialWeeksEmployerCharge>12000.00</PotentialWeeksEmployerCharge>
  </EmployerChargeOccurrence>
  <EmployerChargeOccurrence>
    <BeginWeekNumber>3</BeginWeekNumber>
    <EndWeekNumber>3</EndWeekNumber>
    <TotalWeeklyEmployerCharge>1000.00</TotalWeeklyEmployerCharge>
    <PotentialWeeksEmployerCharge>10000.00</PotentialWeeksEmployerCharge>
  </EmployerChargeOccurrence>
  <WagesInFirstQuarter>9000.00</WagesInFirstQuarter>
  <WagesInSecondQuarter>9000.00</WagesInSecondQuarter>
  <WagesInThirdQuarter>9000.00</WagesInThirdQuarter>
  <WagesInFourthQuarter>9000.00</WagesInFourthQuarter>
  <InitialClaimDate>2018-11-12</InitialClaimDate>
<RequestforReliefofChargesAllowedInd>Y</RequestforReliefofChargesAllowedInd>
</StateMonetaryChargeRequest>
</StateMonetaryChargeRequestCollection>

```

#### 4.4.3.1.1.2 Central Broker Acknowledgement to State Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of requests it received, the number in error, and the dates that it started receiving the records and finished receiving the records. This verifies to the State that the Broker received the desired file so it can move on to the next file.

#### REDACTED

```

<StateRequestFileGUID>D0F7202142A448F0747E99F75CE0FC00</StateRequestFileGUID>
<NumberOfRequestRecordsReceived>63</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2009-07-13T02:37:53.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2009-07-13T02:37:54.000-

```

04:00</DateFinishedReceivingTransmission>

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the “Post” message, the Broker will also return to the State all of the information that it can on why each individual request failed. The FailedMonetaryChargeRequest element defined in the **REDACTED** XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```
<FailedMonetaryChargeRequest>
  <StateMonetaryChargeRequestRecordGUID>000000000000000000000000000099108</StateM
onetaryChargeRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>501</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedMonetaryChargeRequest>
```

Putting this together with the successful acknowledgement:

**REDACTED**

```
<StateRequestFileGUID>D0F7202142A448F0747E99F75CE0FC00</StateRequestFileGUID>
<FailedMonetaryChargeRequest>
  <StateMonetaryChargeRequestRecordGUID>000000000000000000000000000099108</StateM
onetaryChargeRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>501</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedMonetaryChargeRequest>
<FailedMonetaryChargeRequest>
  <StateMonetaryChargeRequestRecordGUID>000000000000000000000000000099935</StateM
onetaryChargeRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>501</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedMonetaryChargeRequest>
<FailedMonetaryChargeRequest>
  <StateMonetaryChargeRequestRecordGUID>000000000000000000000000000099999</StateM
onetaryChargeRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>501</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedMonetaryChargeRequest>
```

```

<FailedMonetaryChargeRequest>
  <StateMonetaryChargeRequestRecordGUID>000000000000000000000000000099998</StateM
onetaryChargeRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>501</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedMonetaryChargeRequest>
<NumberOfRequestRecordsReceived>63</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>4</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2009-07-13T02:37:53.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2009-07-13T02:37:54.000-
04:00</DateFinishedReceivingTransmission>

```

#### 4.4.3.1.2 Integrated Partner Post Payload

##### 4.4.3.1.2.1 Post to Central Broker Payload

The "Post" payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED** xsd

#### **REDACTED**

```

<EmployerTPAMonetaryChargeResponse>
  <StateMonetaryChargeRequestRecordGUID>e755d30ed7dd4662bc0452e9050c0001</StateM
onetaryChargeRequestRecordGUID>
  <BrokerRecordTransactionNumber>641080</BrokerRecordTransactionNumber>
  <ClaimNumber>123456789012345678</ClaimNumber>
  <RequestDate>2019-04-10</RequestDate>
    <ResponseDueDate>2019-05-05</ResponseDueDate>
    <CorrectedFEIN>999999999</CorrectedFEIN>
  <StateEmployerAccountNbr>0123456789</StateEmployerAccountNbr>
  <CorrectedStateEmployerAccountNbr>2123456789</CorrectedStateEmployerAccountNbr>
  <EmployerName>Test Employer</EmployerName>
  <CorrectedEmployerName>Test Employer 2</CorrectedEmployerName>
  <SSN>123000000</SSN>
  <OtherSSN>223001234</OtherSSN>
  <ClaimantLastName>Doe</ClaimantLastName>
  <ClaimantFirstName>Jane</ClaimantFirstName>
  <ClaimantMiddleInitial>A</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>
  <ClaimantNameWorkedAsForEmployer>Donald
Duck</ClaimantNameWorkedAsForEmployer>
    <SeasonalEmploymentInd>Y</SeasonalEmploymentInd>
  <EmployerReportedDependents>2</EmployerReportedDependents>
    <RequestReliefOfChargesInd>N</RequestReliefOfChargesInd>
  <EmployerCorrectedWageforFirstQuarter>1000.0</EmployerCorrectedWageforFirstQuarter>
  <EmployerCorrectedWageforSecondQuarter>1000.0</EmployerCorrectedWageforSecondQu
arter>

```

```

<EmployerCorrectedWageforThirdQuarter>1000.0</EmployerCorrectedWageforThirdQuarter>
<EmployerCorrectedWageforFourthQuarter>1000.0</EmployerCorrectedWageforFourthQuart
er>
<AttachmentOccurrence>
  <DescriptionofAttachmentCode>1</DescriptionofAttachmentCode>
  <TypeofDocument>Test Response Attachment</TypeofDocument>
  <UniqueAttachmentId>01</UniqueAttachmentId>
  <AttachmentSize>1</AttachmentSize>
  <AttachmentData>MA==</AttachmentData>
</AttachmentOccurrence>
<PreparerTypeCode>E</PreparerTypeCode>
<PreparerCompanyName>PreparerCompanyName</PreparerCompanyName>
<PreparerTelephoneNumberPlusExt>2101234567</PreparerTelephoneNumberPlusExt>
<PreparerContactName>Joe Preparer</PreparerContactName>
<PreparerTitle>Preparer Tester</PreparerTitle>
<PreparerFaxNbr>2101234567</PreparerFaxNbr>
<PreparerEmailAddress>prep@asdf.asdf</PreparerEmailAddress>
</EmployerTPAMonetaryChargeResponse>

```

#### 4.4.3.1.2.2 Central Broker Acknowledgement to Integrated Partner Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of responses it received, the number in error, and the dates that it started receiving the records and finished receiving the records. This verifies to the Integrated Partner that the Broker received the desired file so it can move on to the next file.

#### < REDACTED

```

<EmployerTPAResponseFileGUID>542A4AE2395FEDDF1EAA3E57F2DFBCE0</EmployerTPARespo
nseFileGUID>
<NumberOfResponseRecordsReceived>22</NumberOfResponseRecordsReceived>
<NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
<DateStartedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateFinishedReceivingTransmission>

```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the "Post" message, the Broker will also return to the Integrated Partner all of the information that it can on why each individual response failed. The FailedMonetaryChargeResponse element defined in the Monetary Response XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```

<FailedMonetaryChargeResponse>
  <StateMonetaryChargeRequestRecordGUID>000000000000000000000000099943</StateM
onetaryChargeRequestRecordGUID>
  <BrokerRecordTransactionNumber>2001552</BrokerRecordTransactionNumber>
  <ErrorOccurrence>
    <ErrorCode>613</ErrorCode>
    <ErrorMessage>Business Rule violation - The EmployerSeparationReasonCode (Character
– Size 2) must be null if EmployerSepReasonNeededIndictor from the Request is set to "N" for No
and the EmployerSeparationReasonCode does not equal 20 or 21</ErrorMessage>
  </ErrorOccurrence>
</FailedMonetaryChargeResponse>

```

Putting this together with the successful acknowledgement:

### REDACTED

```

<EmployerTPAResponseFileGUID>542A4AE2395FEDDF1EAA3E57F2DFBCE0</EmployerTPARespo
nseFileGUID>

<FailedMonetaryChargeResponse>
  <StateMonetaryChargeRequestRecordGUID>000000000000000000000000099943</StateM
onetaryChargeRequestRecordGUID>
  <BrokerRecordTransactionNumber>2001552</BrokerRecordTransactionNumber>
  <ErrorOccurrence>
<ErrorCode>612</ErrorCode>
    <ErrorMessage> Business Rule violation – The EmployerReportedDependents (Numeric – Size
1) must be null if Dependents for the Request is null.
      </ErrorMessage>
    </ErrorOccurrence>
  </FailedMonetaryChargeResponse>
  <NumberOfResponseRecordsReceived>22</NumberOfResponseRecordsReceived>
  <NumberOfResponseRecordsInError>1</NumberOfResponseRecordsInError>
  <DateStartedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateFinishedReceivingTransmission>
</EmployerTPAMonetaryChargeResponseCollectionAcknowledgement>

```

#### 4.4.3.2 Pull Payload

A “Pull” is defined (see Section 4.1.2) as sending a query to the Central Broker to allow the Central Broker to send the Connector all of the requests or responses it has waiting for it (in multiple transactions if they are from different endpoints or if the data exceeds 8MB). This section discusses the (pre-encryption) payloads in the SOAP message.

##### 4.4.3.2.1 State Pull Payload

###### 4.4.3.2.1.1 Pull from Central Broker

The **REDACTED** defined in the State **REDACTED** WSDL (see Section 4.6) is a complex type that allows the caller to specify one of three operations: a Pull, a Re-Pull by StateSOAPTransactionNumber, and a Re-Pull by a Date Range.

### **REDACTED**

For the regular Pull, the caller needs to supply only the State Unique ID. Although there are different ways to verify the calling State besides this element, the Broker uses it as an additional security check. Also, there is a requirement in the WSDL 1.1 specification that a WSDL definition have at least one input attribute.

### **REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
```

For the Re-Pull by StateSOAPTransactionNumber, the caller needs to supply the state Unique ID and the StateSOAPTransactionNumber element out of the **REDACTED**. This will allow the Broker to send the file defined by the StateSOAPTransactionNumber.

### **REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
<StateMonetaryChargeResponseCollectionQueryCriteria>  
  <StateSOAPTransactionNumber>123456221</StateSOAPTransactionNumber>  
</StateMonetaryChargeResponseCollectionQueryCriteria>
```

For the Re-Pull by date range, the caller needs to supply the State Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and a StateSOAPTransactionNumber.

The first time this operation is called, the StateSOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next StateSOAPTransactionNumber that it sent during that date range (in element name NextStateSOAPTransactionNumber). In the next call to this operation, the caller must include the NextStateSOAPTransactionNumber as the StateSOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.



When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next StateSOAPTransactionNumber (so there will not be a NextStateSOAPTransactionNumber included in the http response SOAP header).

First Call:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
<StateMonetaryChargeResponseCollectionQueryCriteria>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>
</StateMonetaryChargeResponseCollectionQueryCriteria>
```

Subsequent Calls:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
<StateMonetaryChargeResponseCollectionQueryCriteria>
  <StateSOAPTransactionNumber>12345678901234567890123456789012</StateSOAPTransa
ctionNumber>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-
04:00</BrokerRecordEffectiveDateTo>
</StateMonetaryChargeResponseCollectionQueryCriteria>
```

#### 4.4.3.2.1.2 Central Broker Response to the State

When the Broker receives a “Pull” request from a State, it begins assembling all the responses that are intended for that State. It constructs a SOAP message according to the rules for a Monetary & Potential Charge Information SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds the additional field BrokerRecordEffectiveDate to the response, which indicates the date that the response was accepted into the Broker. It then sends the responses in the HTTP response.

**REDACTED**

```
<StateMonetaryChargeRequestRecordGUID>e755d30ed7dd4662bc0452e9050c0001</StateM
onetaryChargeRequestRecordGUID>
<BrokerRecordTransactionNumber>641080</BrokerRecordTransactionNumber>
<ClaimNumber>123456789012345678</ClaimNumber>
<RequestDate>2019-04-10</RequestDate>
  <ResponseDueDate>2019-05-05</ResponseDueDate>
  <CorrectedFEIN>999999999</CorrectedFEIN>
<StateEmployerAccountNbr>0123456789</StateEmployerAccountNbr>
<CorrectedStateEmployerAccountNbr>2123456789</CorrectedStateEmployerAccountNbr>
```

```

<EmployerName>Test Employer</EmployerName>
<CorrectedEmployerName>Test Employer 2</CorrectedEmployerName>
<SSN>123000000</SSN>
<OtherSSN>223001234</OtherSSN>
<ClaimantLastName>Doe</ClaimantLastName>
<ClaimantFirstName>Jane</ClaimantFirstName>
<ClaimantMiddleInitial>A</ClaimantMiddleInitial>
<ClaimantSuffix>Jr</ClaimantSuffix>
<ClaimantNameWorkedAsForEmployer>Donuld
Duck</ClaimantNameWorkedAsForEmployer>
  <SeasonalEmploymentInd>Y</SeasonalEmploymentInd>
  <EmployerReportedDependents>2</EmployerReportedDependents>
  <RequestReliefofChargesInd>N</RequestReliefofChargesInd>
  <EmployerCorrectedWageforFirstQuarter>1000.0</EmployerCorrectedWageforFirstQuarter>
  <EmployerCorrectedWageforSecondQuarter>1000.0</EmployerCorrectedWageforSecondQu
arter>
  <EmployerCorrectedWageforThirdQuarter>1000.0</EmployerCorrectedWageforThirdQuarter>
  <EmployerCorrectedWageforFourthQuarter>1000.0</EmployerCorrectedWageforFourthQuart
er>
  <AttachmentOccurrence>
    <DescriptionofAttachmentCode>1</DescriptionofAttachmentCode>
    <TypeofDocument>Test Response Attachment</TypeofDocument>
    <UniqueAttachmentId>01</UniqueAttachmentId>
    <AttachmentSize>1</AttachmentSize>
    <AttachmentData>MA==</AttachmentData>
  </AttachmentOccurrence>
  <PreparerTypeCode>E</PreparerTypeCode>
  <PreparerCompanyName>PreparerCompanyName</PreparerCompanyName>
  <PreparerTelephoneNumberPlusExt>2101234567</PreparerTelephoneNumberPlusExt>
  <PreparerContactName>Joe Preparer</PreparerContactName>
  <PreparerTitle>Preparer Tester</PreparerTitle>
  <PreparerFaxNbr>2101234567</PreparerFaxNbr>
  <PreparerEmailAddress>prep@asdf.asdf</PreparerEmailAddress>
  <BrokerRecordEffectiveDate>2012-03-20T13:08:34-04:00</BrokerRecordEffectiveDate>
</StateMonetaryChargeResponse>

```

#### 4.4.3.2.1.3 State Acknowledgement to the Central Broker

The **REDACTED** is initiated once the State has received its file from the Broker. The acknowledgement must accompany every State Pull request, as this is the manner in which the Broker knows that the State Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next "Pull" call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the State until the NASWA SIDES team and State Administrator can work out the problem.

The key field in this message is the StateSOAPTransmissionNumber, which must correspond with the StateSOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the state does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

## REDACTED

```
<StateSOAPTransmissionNumber>12345678901234567890123456789012</StateSOAPTransmissionNumber>
  <NumberOfResponseRecordsReceived>0</NumberOfResponseRecordsReceived>
  <NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
  <DateStartedReceivingTransmission>2008-12-31T12:00:00.000-04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2008-12-31T12:00:00.000-04:00</DateFinishedReceivingTransmission>
```

### 4.4.3.2.2 Integrated Partner Pull

#### 4.4.3.2.2.1 Pull from Central Broker

The **REDACTED** is a complex query type that allows the caller to specify one of three operations: a Pull, a Re-Pull by EmployerTPASOAPTransactionNumber, and a Re-Pull by a Date Range.

## REDACTED

For the regular Pull, the caller needs to supply only the Integrated Partner Unique ID. Although there are different ways to verify the calling Integrated Partner besides this element, the Broker uses it as an additional security check. Also, there is a requirement in WSDL 1.1 that a WSDL definition have at least one input attribute.

## REDACTED

```
<UniqueID>BR000000001</UniqueID>
```

For the Re-Pull by EmployerTPASOAPTransactionNumber, the caller needs to supply the Integrated Partner Unique ID and the EmployerTPASOAPTransactionNumber element out of the **REDACTED**. This will allow the Broker to send the file defined by the EmployerTPASOAPTransactionNumber.

## REDACTED

```
<UniqueID>BR000000001</UniqueID>
  <EmployerTPAMonetaryRequestCollectionQueryCriteria>
    <EmployerTPASOAPTransactionNumber>26151</EmployerTPASOAPTransactionNumber>
```

</EmployerTPAMonetaryRequestCollectionQueryCriteria>

For the Re-Pull by date range, the caller needs to supply the Integrated Partner Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex query type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and an EmployerTPASOAPTransactionNumber.

The first time this operation is called, the EmployerTPASOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next EmployerTPASOAPTransactionNumber that it sent during that date range (in element name NextEmployerTPASOAPTransactionNumber). In the next call to this operation, the caller must include the NextEmployerTPASOAPTransactionNumber as the EmployerTPASOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next EmployerTPASOAPTransactionNumber (so there will not be a NextEmployerTPASOAPTransactionNumber included in the http response SOAP header).

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

First Call:

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>  
<EmployerTPAMonetaryRequestCollectionQueryCriteria>  
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-  
04:00</BrokerRecordEffectiveDateFrom>  
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>  
</EmployerTPAMonetaryRequestCollectionQueryCriteria>
```

Subsequent Calls:

## REDACTED

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPAMonetaryRequestCollectionQueryCriteria>
  <EmployerTPASOAPTransactionNumber>12345678901234567890123456789012</Employer
TPASOAPTransactionNumber>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-
04:00</BrokerRecordEffectiveDateTo>
```

### 4.4.3.2.2 Central Broker Response to Integrated Partner

When the Broker receives a “Pull” request from an Integrated Partner, it begins assembling all the requests that are intended for that Integrated Partner. It constructs a SOAP message according to the rules for a Monetary & Potential Charges SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds two additional fields to the Monetary & Potential Charges Request - the BrokerRecordEffectiveDate and the BrokerRecordTransactionNumber. The BrokerRecordEffectiveDate indicates the date that the request was accepted into the Broker. The BrokerRecordTransactionNumber is a unique record tracking number and must be returned on the response for this record. It then sends the Monetary & Potential Charges requests in the HTTP response.

## REDACTED

```
<StateMonetaryChargeRequestRecordGUID>e755d30ed7dd4662bc0452e9050c0001</State
MonetaryChargeRequestRecordGUID>
  <ClaimNumber>123456789012345678</ClaimNumber>
  <FormNumber>form12345678</FormNumber>
  <RequestDate>2019-04-10</RequestDate>
  <ResponseDueDate>2019-05-05</ResponseDueDate>
  <RequestingStateAbbreviation>CA</RequestingStateAbbreviation>
  <UIOfficeName>CA UIOfficeName</UIOfficeName>
  <UIOfficePhone>2101234567</UIOfficePhone>
  <UIOfficeFax>2101234568</UIOfficeFax>
  <UIOfficeEmailAddress>UIOfficeEmailAddress@asdf.gov</UIOfficeEmailAddress>
  <FEIN>999999999</FEIN>
  <StateEmployerAccountNbr>0123456789</StateEmployerAccountNbr>
  <PredecessorEmployerAccountNbr>0123456789</PredecessorEmployerAccountNbr>
  <EmployerName>Test Employer</EmployerName>
  <EmployerAccountType>1</EmployerAccountType>
  <TypeofEmployerCode>1</TypeofEmployerCode>
  <SSN>123000000</SSN>
  <ClaimantLastName>Doe</ClaimantLastName>
  <OtherLastName>Mouse</OtherLastName>
  <ClaimantFirstName>Jane</ClaimantFirstName>
  <ClaimantMiddleInitial>A</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>
  <Dependents>2</Dependents>
  <ClaimantSepReasonCode>2</ClaimantSepReasonCode>
```

```

<BenefitYearBeginDate>2013-07-23</BenefitYearBeginDate>
<BasePeriodStartDate>2010-07-24</BasePeriodStartDate>
<BasePeriodEndDate>2011-07-23</BasePeriodEndDate>
<NumberEmployersInBasePeriod>1</NumberEmployersInBasePeriod>
<BasePeriodWagesAllEmployers>42000.00</BasePeriodWagesAllEmployers>
<BasePeriodWagesThisEmployer>36000.00</BasePeriodWagesThisEmployer>
<WeeklyBenefitAmount>1000.00</WeeklyBenefitAmount>
<MaximumBenefitAmount>60000.00</MaximumBenefitAmount>
<TotalWeeksOfBenefits>52</TotalWeeksOfBenefits>
<PercentChargableEmployer>100.00</PercentChargableEmployer>
<TotalEmployerCharge>50000.00</TotalEmployerCharge>
  <EmployerChargeOccurrence>
    <BeginWeekNumber>1</BeginWeekNumber>
    <EndWeekNumber>1</EndWeekNumber>
    <TotalWeeklyEmployerCharge>1000.00</TotalWeeklyEmployerCharge>
    <PotentialWeeksEmployerCharge>10000.00</PotentialWeeksEmployerCharge>
  </EmployerChargeOccurrence>
  <EmployerChargeOccurrence>
    <BeginWeekNumber>2</BeginWeekNumber>
    <EndWeekNumber>2</EndWeekNumber>
    <TotalWeeklyEmployerCharge>1200.00</TotalWeeklyEmployerCharge>
    <PotentialWeeksEmployerCharge>12000.00</PotentialWeeksEmployerCharge>
  </EmployerChargeOccurrence>
  <EmployerChargeOccurrence>
    <BeginWeekNumber>3</BeginWeekNumber>
    <EndWeekNumber>3</EndWeekNumber>
    <TotalWeeklyEmployerCharge>1000.00</TotalWeeklyEmployerCharge>
    <PotentialWeeksEmployerCharge>10000.00</PotentialWeeksEmployerCharge>
  </EmployerChargeOccurrence>
<WagesInFirstQuarter>9000.00</WagesInFirstQuarter>
<WagesInSecondQuarter>9000.00</WagesInSecondQuarter>
<WagesInThirdQuarter>9000.00</WagesInThirdQuarter>
<WagesInFourthQuarter>9000.00</WagesInFourthQuarter>
<InitialClaimDate>2018-11-12</InitialClaimDate>
<RequestforReliefofChargesAllowedInd>Y</RequestforReliefofChargesAllowedInd>
  <BrokerRecordTransactionNumber>261089</BrokerRecordTransactionNumber>
  <BrokerRecordEffectiveDate>2012-02-22T13:43:48-05:00</BrokerRecordEffectiveDate>
</EmployerTPAMonetaryChargeRequest>

```

#### 4.4.3.2.2.3 Integrated Partner Acknowledgement to Central Broker

The **REDACTED** is initiated once the Integrated Partner has received its file from the Broker. The acknowledgement must accompany every Integrated Partner Pull request, as this is the manner in which the Broker knows that the Integrated Partner Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next "Pull" call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will

suspend any processing of Pull requests by the Integrated Partner until the NASWA SIDES team and Integrated Partner Administrator can work out the problem.

The key field in this message is the EmployerTPASOAPTransmissionNumber, which must correspond with the EmployerTPASOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the Integrated Partner does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

#### **REDACTED**

```
<EmployerTPASOAPTransmissionNumber>3211</EmployerTPASOAPTransmissionNumber>
<NumberOfRequestRecordsReceived>4</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateFinishedReceivingTransmission>
```

### **4.4.4 Determinations & Decisions**

#### **4.4.4.1 Post Payload**

A “Post” is defined (see Section 4.1.1) as sending a request or response to the Broker by a particular Connector. This section discusses the (pre-encryption) payloads in the SOAP message.

##### **4.4.4.1.1 State Post Payload**

###### **4.4.4.1.1.1 Post to Central Broker Payload**

The “Post” payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED**.xsd.

#### **REDACTED**

```
<StateDeterminationDecisionRequest>
<StateDeterminationDecisionRequestRecordGUID>1213000000000000000000000000000011111</StateD
eterminationDecisionRequestRecordGUID>
  <RequestDate>2014-06-15</RequestDate>
  <RequestingStateAbbreviation>ST</RequestingStateAbbreviation>
  <SSN>011311111</SSN>
  <ClaimantLastName>Doe</ClaimantLastName>
  <OtherLastName>Simpson</OtherLastName>
  <ClaimantFirstName>Homer</ClaimantFirstName>
  <ClaimEffectiveDate>2013-07-23</ClaimEffectiveDate>
  <ClaimNumber>99991111999919</ClaimNumber>
  <BenefitYearBeginDate>2013-05-23</BenefitYearBeginDate>
  <StateEmployerAccountNbr>123456789</StateEmployerAccountNbr>
  <FEIN>123000656</FEIN>
```

```

<EmployerName>JC Penney Inc</EmployerName>
<UIOfficeName>CA UI Claims</UIOfficeName>
<DeterminationOrDecisionInd>12</DeterminationOrDecisionInd>
<DeterminationOrDecisionDate>2013-09-20</DeterminationOrDecisionDate>
<DeterminationOrDecisionResult>11</DeterminationOrDecisionResult>
<EmployerChargableForBenefitsPaid>Y</EmployerChargableForBenefitsPaid>
<AgentResponseIndicator>1</AgentResponseIndicator>
<EmpWorksiteIndicator>1</EmpWorksiteIndicator>
<NotAvailableIndicator>1</NotAvailableIndicator>
<InterpreterIndicator>1</InterpreterIndicator>
<SpecialNeedsIndicator>1</SpecialNeedsIndicator>
<AttachmentOccurrence>
  <DescriptionofAttachmentCode>1</DescriptionofAttachmentCode>
  <TypeofDocument>Test Request Attachment</TypeofDocument>
  <UniqueAttachmentId>01</UniqueAttachmentId>
  <ActionableAttachment>11</ActionableAttachment>
  <AttachmentSize>1</AttachmentSize>
  <AttachmentData>e1xydGYxXGFkSANClxwYXlGfX0=</AttachmentData>
</AttachmentOccurrence>
<AppealableDeterminationOrDecision>1</AppealableDeterminationOrDecision>
<LastDateToAppeal>2014-07-02</LastDateToAppeal>
</StateDeterminationDecisionRequest>

```

#### 4.4.4.1.1.2 Central Broker Acknowledgement to State Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of requests it received, the number in error, and the dates that it started receiving the records and finished receiving the records. This verifies to the State that the Broker received the desired file so it can move on to the next file.

#### REDACTED

```

<StateRequestFileGUID>D0F7202142A448F0747E99F75CE0FC00</StateRequestFileGUID>
<NumberOfRequestRecordsReceived>63</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2009-07-13T02:37:53.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2009-07-13T02:37:54.000-
04:00</DateFinishedReceivingTransmission>

```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the “Post” message, the Broker will also return to the State all of the information that it can on why each individual request failed. The **REDACTED** element defined in the Determination Decision XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.



```
<FailedDeterminationDecisionRequest>
  <StateDeterminationDecisionRequestRecordGUID>000000000000000000000000099108</StateDeterminationDecisionRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>701</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedDeterminationDecisionRequest>
```

Putting this together with the successful acknowledgement:

**REDACTED**

```
<StateRequestFileGUID>D0F7202142A448F0747E99F75CE0FC00</StateRequestFileGUID>
<FailedDeterminationDecisionRequest>
  <StateDeterminationDecisionRequestRecordGUID>000000000000000000000000099108</StateDeterminationDecisionRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>701</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedDeterminationDecisionRequest>
<FailedDeterminationDecisionRequest>
  <StateDeterminationDecisionRequestRecordGUID>000000000000000000000000099935</StateDeterminationDecisionRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>701</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedDeterminationDecisionRequest>
<FailedDeterminationDecisionRequest>
  <StateDeterminationDecisionRequestRecordGUID>000000000000000000000000099999</StateDeterminationDecisionRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>701</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedDeterminationDecisionRequest>
<FailedDeterminationDecisionRequest>
  <StateDeterminationDecisionRequestRecordGUID>000000000000000000000000099998</StateDeterminationDecisionRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>701</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedDeterminationDecisionRequest>
<NumberOfRequestRecordsReceived>63</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>4</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2009-07-13T02:37:53.000-04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2009-07-13T02:37:54.000-04:00</DateFinishedReceivingTransmission>
```



```

    <PreparerContactName>Dean Smity</PreparerContactName>
    <PreparerTitle>Manager</PreparerTitle>
    <PreparerEmailAddress>snooty@snoots.com</PreparerEmailAddress>
  </EmployerTPADeterminationDecisionResponse>

```

#### 4.4.4.1.2.2 Central Broker Acknowledgement to Integrated Partner Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of responses it received, the number in error, and the dates that it started receiving the records and finished receiving the records. This verifies to the Integrated Partner that the Broker received the desired file so it can move on to the next file.

#### REDACTED

```

  <EmployerTPAResponseFileGUID>542A4AE2395FEDDF1EAA3E57F2DFBCE0</EmployerTPAResponseFileGUID>
  <NumberOfResponseRecordsReceived>22</NumberOfResponseRecordsReceived>
  <NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
  <DateStartedReceivingTransmission>2009-07-22T03:13:50.000-04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2009-07-22T03:13:50.000-04:00</DateFinishedReceivingTransmission>

```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the “Post” message, the Broker will also return to the Integrated Partner all of the information that it can on why each individual response failed. The **REDACTED** element defined in the Determination Decisions Response XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```

  <FailedDeterminationDecisionResponse>
    <StateDeterminationDecisionRequestRecordGUID>000000000000000000000000099943</StateDeterminationDecisionRequestRecordGUID>
    <BrokerRecordTransactionNumber>2001552</BrokerRecordTransactionNumber>
    <ErrorOccurrence>
      <ErrorCode>813</ErrorCode>
      <ErrorMessage>Business Rule violation - The AppealReasonIndicator (Character – Size 2) required if EmployerRequestingRedetOrAppealOfDeterminationOrDecision is equal to “Y” otherwise it must be null.</ErrorMessage>
    </ErrorOccurrence>
  </FailedMonetaryChargeResponse>

```

Putting this together with the successful acknowledgement:

## REDACTED

```
<EmployerTPAResponseFileGUID>542A4AE2395FEDDF1EAA3E57F2DFBCE0</EmployerTPAResponseFileGUID>

<FailedDeterminationDecisionResponse>
  <StateDeterminationDecisionRequestRecordGUID>000000000000000000000000000099943</StateDeterminationDecisionRequestRecordGUID>
  <BrokerRecordTransactionNumber>2001552</BrokerRecordTransactionNumber>
  <ErrorOccurrence>
    <ErrorCode>813</ErrorCode>
    <ErrorMessage>Business Rule violation - The AppealReasonIndicator (Character – Size 2) required if EmployerRequestingRedetOrAppealOfDeterminationOrDecision is equal to “Y” otherwise it must be null.</ErrorMessage>
  </ErrorOccurrence>
</FailedDeterminationDecisionResponse>
<NumberOfResponseRecordsReceived>22</NumberOfResponseRecordsReceived>
<NumberOfResponseRecordsInError>1</NumberOfResponseRecordsInError>
<DateStartedReceivingTransmission>2009-07-22T03:13:50.000-04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2009-07-22T03:13:50.000-04:00</DateFinishedReceivingTransmission>
```

### 4.4.4.2 Pull Payload

A “Pull” is defined (see Section 4.1.2) as sending a query to the Central Broker to allow the Central Broker to send the Connector all of the requests or responses it has waiting for it (in multiple transactions if they are from different endpoints or if the data exceeds 8MB). This section discusses the (pre-encryption) payloads in the SOAP message.

#### 4.4.4.2.1 State Pull Payload

##### 4.4.4.2.1.1 Pull from Central Broker

The **REDACTED** defined in the State Determination Decision WSDL (see Section 4.6) is a complex type that allows the caller to specify one of three operations: a Pull, a Re-Pull by StateSOAPTransactionNumber, and a Re-Pull by a Date Range.

```
<!-- Query element for states to collect claim responses they are expecting -->
```

## REDACTED

For the regular Pull, the caller needs to supply only the State Unique ID. Although there are different ways to verify the calling State besides this element, the Broker uses it as an additional security check. Also, there is a requirement in the WSDL 1.1 specification that a WSDL definition have at least one input attribute.

## REDACTED

```
<StatePostalCode>ST</StatePostalCode>
```

For the Re-Pull by StateSOAPTransactionNumber, the caller needs to supply the State Unique ID and the StateSOAPTransactionNumber element out of the StateDeterminationDecisionResponseCollectionQueryCriteriaType. This will allow the Broker to send the file defined by the StateSOAPTransactionNumber.

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
<StateDeterminationDecisionResponseCollectionQueryCriteria>  
  <StateSOAPTransactionNumber>123456221</StateSOAPTransactionNumber>  
</StateDeterminationDecisionResponseCollectionQueryCriteria>
```

For the Re-Pull by date range, the caller needs to supply the State Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and a StateSOAPTransactionNumber.

The first time this operation is called, the StateSOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next StateSOAPTransactionNumber that it sent during that date range (in element name NextStateSOAPTransactionNumber). In the next call to this operation, the caller must include the NextStateSOAPTransactionNumber as the StateSOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next StateSOAPTransactionNumber (so there will not be a NextStateSOAPTransactionNumber included in the http response SOAP header).

First Call:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
<StateDeterminationDecisionResponseCollectionQueryCriteria>  
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-  
04:00</BrokerRecordEffectiveDateFrom>
```



```

comments</EmployerReasonRequestingRedetOrAppeal>
  <AttachmentOccurrence>
    <DescriptionofAttachmentCode>1</DescriptionofAttachmentCode>
    <TypeofDocument>Test Request Attachment</TypeofDocument>
    <UniqueAttachmentId>01</UniqueAttachmentId>
    <AttachmentSize>1</AttachmentSize>
    <AttachmentData>e1xydGYxXGFkZWZsYW5nMTAyNVxhbnNpXGXlglfX0=</AttachmentData>
  </AttachmentOccurrence>
  <AgentType>2</AgentType>
  <AgentAttorneyName>Roger Smith</AgentAttorneyName>
  <AgentAttorneyAddress1>1 Penn State Ave</AgentAttorneyAddress1>
  <AgentAttorneyAddress2>#100</AgentAttorneyAddress2>
  <AgentAttorneyCity>Philadelphia</AgentAttorneyCity>
  <AgentAttorneyState>PA</AgentAttorneyState>
  <AgentAttorneyZip>06877</AgentAttorneyZip>

<AgentAttorneyTelephoneNumberPlusExt>2132122122</AgentAttorneyTelephoneNumberPlusExt>
>
  <InterpreterLanguage>Interpret in Spanish</InterpreterLanguage>
  <SpecialNeeds>No Special Needs Required</SpecialNeeds>
  <PreparerTypeCode>E</PreparerTypeCode>
  <PreparerTelephoneNumberPlusExt>2132122122</PreparerTelephoneNumberPlusExt>
  <PreparerContactName>Dean Smity</PreparerContactName>
  <PreparerTitle>Manager</PreparerTitle>
  <PreparerEmailAddress>snooty@snoots.com</PreparerEmailAddress>
  <BrokerRecordEffectiveDate>2014-04-03T07:11:07-04:00</BrokerRecordEffectiveDate>
</StateDeterminationDecisionResponse>

```

#### 4.4.4.2.1.3 State Acknowledgement to the Central Broker

The **REDACTED** is initiated once the State has received its file from the Broker. The acknowledgement must accompany every State Pull request, as this is the manner in which the Broker knows that the State Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next “Pull” call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the State until the NASWA SIDES team and State Administrator can work out the problem.

The key field in this message is the StateSOAPTransmissionNumber, which must correspond with the StateSOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the State does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

## REDACTED

```
<StateSOAPTransmissionNumber>12345678901234567890123456789012</StateSOAPTransmissionNumber>
  <NumberOfResponseRecordsReceived>0</NumberOfResponseRecordsReceived>
  <NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
  <DateStartedReceivingTransmission>2008-12-31T12:00:00.000-04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2008-12-31T12:00:00.000-04:00</DateFinishedReceivingTransmission>
```

### 4.4.4.2.2 Integrated Partner Pull

#### 4.4.4.2.2.1 Pull from Central Broker

The **REDACTED** is a complex query type that allows the caller to specify one of three operations: a Pull, a Re-Pull by EmployerTPASOAPTransactionNumber, and a Re-Pull by a Date Range.

*<!-- Query element for employer to collect claim responses they are expecting -->*

#### REDACTED

For the regular Pull, the caller needs to supply only the Integrated Partner Unique ID. Although there are different ways to verify the calling Integrated Partner besides this element, the Broker uses it as an additional security check. Also, there is a requirement in WSDL 1.1 that a WSDL definition have at least one input attribute.

#### REDACTED

```
<UniqueID>BR000000001</UniqueID>
```

For the Re-Pull by EmployerTPASOAPTransactionNumber, the caller needs to supply the Integrated Partner Unique ID and the EmployerTPASOAPTransactionNumber element out of the **REDACTED**. This will allow the Broker to send the file defined by the EmployerTPASOAPTransactionNumber.

#### REDACTED

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPADeterminationDecisionRequestCollectionQueryCriteria>
  <EmployerTPASOAPTransactionNumber>26151</EmployerTPASOAPTransactionNumber>
</EmployerTPADeterminationDecisionRequestCollectionQueryCriteria>
```

For the Re-Pull by date range, the caller needs to supply the Integrated Partner Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.



The **REDACTED** is a complex query type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and an EmployerTPASOAPTransactionNumber.

The first time this operation is called, the EmployerTPASOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next EmployerTPASOAPTransactionNumber that it sent during that date range (in element name NextEmployerTPASOAPTransactionNumber). In the next call to this operation, the caller must include the NextEmployerTPASOAPTransactionNumber as the EmployerTPASOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next EmployerTPASOAPTransactionNumber (so there will not be a NextEmployerTPASOAPTransactionNumber included in the http response SOAP header).

First Call:

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPADeterminationDecisionRequestCollectionQueryCriteria>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>
</EmployerTPADeterminationDecisionRequestCollectionQueryCriteria>
```

Subsequent Calls:

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPADeterminationDecisionRequestCollectionQueryCriteria>
  <EmployerTPASOAPTransactionNumber>12345678901234567890123456789012</Employer
TPASOAPTransactionNumber>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-
04:00</BrokerRecordEffectiveDateTo>
```

</EmployerTPADeterminationDecisionRequestCollectionQueryCriteria>

#### 4.4.4.2.2.2 Central Broker Response to Integrated Partner

When the Broker receives a “Pull” request from an Integrated Partner, it begins assembling all the requests that are intended for that Integrated Partner. It constructs a SOAP message according to the rules for a Determinations & Decisions Information SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds two additional fields to the Determinations & Decisions Request - the BrokerRecordEffectiveDate and the BrokerRecordTransactionNumber. The BrokerRecordEffectiveDate indicates the date that the request was accepted into the Broker. The BrokerRecordTransactionNumber is a unique record tracking number and must be returned on the response for this record. It then sends the Determinations & Decisions Charges requests in the HTTP response.

#### REDACTED

```
<StateDeterminationDecisionRequestRecordGUID>121371600000000000000000000011111<
/StateDeterminationDecisionRequestRecordGUID>
  <RequestDate>2014-06-15</RequestDate>
  <RequestingStateAbbreviation>ST</RequestingStateAbbreviation>
  <SSN>011311111</SSN>
  <ClaimantLastName>Doe</ClaimantLastName>
  <OtherLastName>Simpson</OtherLastName>
  <ClaimantFirstName>Homer</ClaimantFirstName>
  <ClaimEffectiveDate>2013-07-23</ClaimEffectiveDate>
  <ClaimNumber>99991111999919</ClaimNumber>
  <BenefitYearBeginDate>2013-05-23</BenefitYearBeginDate>
  <StateEmployerAccountNbr>123456789</StateEmployerAccountNbr>
  <FEIN>123000656</FEIN>
  <EmployerName>JC Penney Inc</EmployerName>
  <UIOfficeName>CA UI Claims</UIOfficeName>
  <DeterminationOrDecisionInd>30</DeterminationOrDecisionInd>
  <DeterminationOrDecisionDate>2013-09-20</DeterminationOrDecisionDate>
  <DeterminationOrDecisionResult>11</DeterminationOrDecisionResult>
  <EmployerChargableForBenefitsPaid>Y</EmployerChargableForBenefitsPaid>
  <AgentResponseIndicator>1</AgentResponseIndicator>
  <EmpWorksiteIndicator>1</EmpWorksiteIndicator>
  <NotAvailableIndicator>1</NotAvailableIndicator>
  <InterpreterIndicator>1</InterpreterIndicator>
  <SpecialNeedsIndicator>1</SpecialNeedsIndicator>
  <AttachmentOccurrence>
    <DescriptionofAttachmentCode>1</DescriptionofAttachmentCode>
    <TypeofDocument>Test Request Attachment</TypeofDocument>
    <UniqueAttachmentId>01</UniqueAttachmentId>
    <ActionableAttachment>11</ActionableAttachment>
    <AttachmentSize>1</AttachmentSize>
    <AttachmentData>e1xydGYxXGFkZWZsYW5XlGfX0=</AttachmentData>
  </AttachmentOccurrence>
```

```
<AppealableDeterminationOrDecision>1</AppealableDeterminationOrDecision>
<LastDateToAppeal>2014-07-02</LastDateToAppeal>
<BrokerRecordTransactionNumber>8602</BrokerRecordTransactionNumber>
<BrokerRecordEffectiveDate>2014-02-07T13:21:06-
05:00</BrokerRecordEffectiveDate>
</EmployerTPADeterminationDecisionRequest>
```

#### 4.4.4.2.2.3 Integrated Partner Acknowledgement to Central Broker

The **REDACTED** is initiated once the Integrated Partner has received its file from the Broker. The acknowledgement must accompany every Integrated Partner Pull request, as this is the manner in which the Broker knows that the Integrated Partner Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next "Pull" call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the Integrated Partner until the NASWA SIDES team and Integrated Partner Administrator can work out the problem.

The key field in this message is the EmployerTPASOAPTransmissionNumber, which must correspond with the EmployerTPASOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the Integrated Partner does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

#### **REDACTED**

```
<EmployerTPASOAPTransmissionNumber>3211</EmployerTPASOAPTransmissionNumber>
<NumberOfRequestRecordsReceived>4</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateFinishedReceivingTransmission>
```

### 4.4.5 Benefit Charges

#### 4.4.5.1 Post Payload

A "Post" is defined (see Section 4.1.1) as sending a request or response to the Broker by a particular Connector. This section discusses the (pre-encryption) payloads in the SOAP message.

##### 4.4.5.1.1 State Post Payload

##### 4.4.5.1.1.1 Post to Central Broker Payload

The "Post" payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED** xsd.

**REDACTED**

```
<StateBenefitChargesRequest>
  <!--Required: 32 characters-->

  <StateChargeRecordGUID>00000000000000000000000000000802</StateChargeR
  ecordGUID>
  <!--Required: yyyy-mm-dd: Must be previous date, or less than current date + 6
  months-->
  <NoticeDate>2016-11-08</NoticeDate>
  <!--Required: Y = Yes, N = No-->
  <DueDateRequiredInd>Y</DueDateRequiredInd>
  <!--Conditional: yyyy-mm-dd: Required if DueDateRequiredInd is Y, -->
  <LastDayToRespond>2016-11-29</LastDayToRespond><!--Required: 2 Characters-
  -->
  <ChargingStateAbbreviation>ST</ChargingStateAbbreviation><!--Required: Up
  to 25 characters-->
  <UIOfficeName>Test Office 2</UIOfficeName><!--Optional: up to 60 characters--
  >
  <UIOfficeEmailAddress>test@email.com</UIOfficeEmailAddress><!--Optional: Up
  to 15 digits-->
  <UIOfficePhone>3014310826</UIOfficePhone>
  <!--Optional: Up to 10 digits-->
  <UIOfficeFax>3014310827</UIOfficeFax><!--Required: 1 Digit: 1 = Reserve Ratio, 2
  = Benefit Ratio (may also have a reserve ratio component)--><!--3 = Benefit Wage Ratio,
  4 = Payroll Variation-->
  <ExperienceRatingMethod>2</ExperienceRatingMethod>
  <!--Required: Up to 20 characters-->
  <StateEmployerAccountNbr>8000001</StateEmployerAccountNbr>
  <!--Required: Up to 100 characters-->
  <EmployerName>Second Test Company</EmployerName>
  <!--Required: 9 Digits-->
  <FEIN>301416817</FEIN>
  <!--Required: yyyy-mm-dd: Must be less than or equal to NoticeDate minus seven
  days and equal to or greater than NoticeDate minus 13 months, 7 days-->
  <ChargeStatementStartDate>2016-11-01</ChargeStatementStartDate>
  <!--Required: yyyy-mm-dd: Greater or equal to 7, and less than 93 days after
  StartDate-->
  <ChargeStatementEndDate>2016-11-29</ChargeStatementEndDate>
  <!--Required: 1 = taxable employer, 2 = reimbursable employer-->
  <EmployerAccountType>1</EmployerAccountType>
  <!--Required: Up to 7 numbers: 1 or larger-->
  <NumberOfIndividualSSNsWithCharges>2</NumberOfIndividualSSNsWithCharges
  >
  <!--Required: 15 Number, 2 decimals automatic-->
  <TotalDollarAmountChargedToEmployer>5400.00</TotalDollarAmountChargedTo
  Employer>

  <!--Part 2: Required: Repeatable 1-n times, n =
  NumberOfIndividualSSNsWithCharges-->
```

<SSNBYBChargeProgramCode><!--Required: 9 numbers-->  
 <SSN>216262448</SSN>  
 <!--Required: yyyy-mm-dd: At least 7 days prior to NoticeDate, not more than 6 years before NoticeDate-->  
 <BenefitYearBeginDate>2016-09-19</BenefitYearBeginDate><!--  
 Conditional: Required for each SSNBYBChargeProgramCode: 1 Number: 1 = Regular State UI Benefits--><!--2 = Combined Wage Claim, 3 = Federal Extended Benefits, 4 = State Extended Benefits, 5 = Combined Wage Claim extended benefits-->  
 <ProgramCode>1</ProgramCode>  
 <!--Optional: Up to 18 characters-->  
 <ClaimNumber>802</ClaimNumber>  
 <!--Required: Up to 30 characters-->  
 <ClaimantLastName>Adams</ClaimantLastName>  
 <!--Optional: Up to 30 characters-->  
 <OtherLastName>Smith</OtherLastName>  
 <!--Required: Up to 25 characters-->  
 <ClaimantFirstName>John</ClaimantFirstName>  
 <!--Required: 1 character-->  
 <ClaimantMiddleInitial>A</ClaimantMiddleInitial>  
 <!--Optional: 1 number: 1 = Last, Base period, and Lag quarter employer, 2 = Last employer, 3 = Base period employer, 4 = Lag quarter employer, 5 = Last and Base Period employer, 6 = Last and Lag Quarter employer, 7 = Base period and Lag quarter employer-->  
 <TypeofEmployerCode>5</TypeofEmployerCode>  
 <!--Required: yyyy-mm-dd: Must be equal to or more recent than BenefitYearBeginDate minus 79 weeks, and at least 52 weeks prior to BenefitYearBeginDate-->  
 <BasePeriodStartDate>2015-07-07</BasePeriodStartDate>  
 <!--Required: yyyy-mm-dd: Must equal BasePeriodStartDate plus 53 weeks-->  
 <BasePeriodEndDate>2016-07-12</BasePeriodEndDate>  
 <!--Optional: Up to 15 numbers, 2 decimals automatic-->  
 <BasePeriodWages>42807.69</BasePeriodWages>  
 <!--Required: Up to 15 numbers, 2 decimals: Negative numbers are for credits-->  
 <TotalAmountChargedToEmployer>2700</TotalAmountChargedToEmployer>  
 <!--Required: Up to 2 numbers: 1 - 33-->  
 <TotalNumberOfWeeksChargedToEmployerAccount>9</TotalNumberOfWeeksChargedToEmployerAccount>  
 <!--Optional: Up to 2000 characters-->  
 <BenefitChargeComment>Charged at 300.00 per week</BenefitChargeComment>  
 </SSNBYBChargeProgramCode>  
 <!--Part 2: Required: Repeatable 1-n times, n = NumberOfIndividualSSNsWithCharges-->  
 <SSNBYBChargeProgramCode><!--Required: 9 numbers-->  
 <SSN>498653795</SSN>  
 <!--Required: yyyy-mm-dd: At least 7 days prior to NoticeDate, not more than 6 years before NoticeDate-->  
 <BenefitYearBeginDate>2016-09-19</BenefitYearBeginDate><!--  
 Conditional: Required for each SSNBYBChargeProgramCode: 1 Number: 1 = Regular State UI

Benefits--><!--2 = Combined Wage Claim, 3 = Federal Extended Benefits, 4 = State Extended Benefits, 5 = Combined Wage Claim extended benefits-->  
 <ProgramCode>2</ProgramCode>  
 <!--Optional: Up to 18 characters-->  
 <ClaimNumber>1802</ClaimNumber>  
 <!--Required: Up to 30 characters-->  
 <ClaimantLastName>Grey</ClaimantLastName>  
 <!--Required: Up to 25 characters-->  
 <ClaimantFirstName>Patrick</ClaimantFirstName>  
 <!--Required: 1 character-->  
 <ClaimantMiddleInitial>J</ClaimantMiddleInitial>  
 <!--Optional: Up to 8 characters-->  
 <ClaimantSuffix>Jr</ClaimantSuffix>  
 <!--Optional: 1 number: 1 = Last, Base period, and Lag quarter employer, 2 = Last employer, 3 = Base period employer--><!--4 = Lag quarter employer, 5 = Last and Base Period employer, 6 = Last and Lag Quarter employer, 7 = Base period and Lag quarter employer-->  
 <TypeofEmployerCode>1</TypeofEmployerCode>  
 <!--Required: yyyy-mm-dd: Must be equal to or more recent than BenefitYearBeginDate minus 79 weeks, and at least 52 weeks prior to BenefitYearBeginDate-->  
 <BasePeriodStartDate>2015-07-07</BasePeriodStartDate>  
 <!--yyyy-mm-dd: Must equal BasePeriodStartDate plus 53 weeks-->  
 <BasePeriodEndDate>2016-07-12</BasePeriodEndDate>  
 <!--Required: Up to 15 numbers, 2 decimals: Negative numbers are for credits-->  
 <TotalAmountChargedToEmployer>2700</TotalAmountChargedToEmployer>  
 <!--Required: Up to 2 numbers: 1 - 33-->  
 <TotalNumberOfWeeksChargedToEmployerAccount>6</TotalNumberOfWeeksChargedToEmployerAccount>  
 <!--Optional: Up to 2000 characters-->  
 <BenefitChargeComment>450.00 per week</BenefitChargeComment>  
 </SSNBYBChargeProgramCode>  
 <!--Required: 1 number: 1 = Yes, 2 = No-->  
 <AgentResponseIndicator>2</AgentResponseIndicator>  
 <!--Required: 1 number: 1 = Yes, 2 = No-->  
 <NotAvailableIndicator>2</NotAvailableIndicator>  
 <!--Required: 1 number: 1 = Yes, 2 = No-->  
 <InterpreterIndicator>2</InterpreterIndicator>  
 <!--Required: 1 number: 1 = Yes, 2 = No-->  
 <SpecialNeedsIndicator>2</SpecialNeedsIndicator>  
 <!--Required: 1 number: 1 = Protest, 2 = Appeal, 3 = Question, 4 = Protest/Question, 5 = Appeal/Question-->  
 <StateTerminologyUsed>1</StateTerminologyUsed>  
 </StateBenefitChargesRequest>

#### 4.4.5.1.1.2 Central Broker Acknowledgement to State Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of requests it received, the number in error, and the dates that it started receiving the records and finished receiving the records. This

verifies to the State that the Broker received the desired file so it can move on to the next file.

```
REDACTED <StateRequestFileGUID>D0F7202142A448F0747E99F75CE0FC00</StateRequestFileGUID>
  <NumberOfRequestRecordsReceived>63</NumberOfRequestRecordsReceived>
  <NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
  <DateStartedReceivingTransmission>2009-07-13T02:37:53.000-04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2009-07-13T02:37:54.000-04:00</DateFinishedReceivingTransmission>
```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the “Post” message, the Broker will also return to the State all of the information that it can on why each individual request failed. The FailedBenefitChargesRequest element defined in the Benefit Charges XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```
<FailedBenefitChargesRequest>
  <StateBenefitChargesRequestRecordGUID>000000000000000000000000099108</StateBenefitChargesRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>701</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedBenefitChargesRequest>
```

Putting this together with the successful acknowledgement:

```
REDACTED
  <StateRequestFileGUID>D0F7202142A448F0747E99F75CE0FC00</StateRequestFileGUID>
  <FailedBenefitChargesRequest>
    <StateBenefitChargesRequestRecordGUID>000000000000000000000000099108</StateBenefitChargesRequestRecordGUID>
    <ErrorOccurrence>
      <ErrorCode>901</ErrorCode>
      <ErrorMessage>XSD validation violation</ErrorMessage>
    </ErrorOccurrence>
  </FailedBenefitChargesRequest>
  <FailedBenefitChargesRequest>
    <StateBenefitChargesRequestRecordGUID>000000000000000000000000099935</StateBenefitChargesRequestRecordGUID>
    <ErrorOccurrence>
      <ErrorCode>901</ErrorCode>
      <ErrorMessage>XSD validation violation</ErrorMessage>
```

```

    </ErrorOccurrence>
  </FailedBenefitChargesRequest>
<FailedBenefitChargesRequest>
  <StateBenefitChargesRequestRecordGUID>000000000000000000000000099999</StateBen
efitChargesRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>901</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedBenefitChargesRequest>
<FailedBenefitChargesRequest>
  <StateBenefitChargesRequestRecordGUID>000000000000000000000000099998</StateMo
netaryChargeRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>901</ErrorCode>
    <ErrorMessage>XSD validation violation</ErrorMessage>
  </ErrorOccurrence>
</FailedBenefitChargesRequest>
<NumberOfRequestRecordsReceived>63</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>4</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2009-07-13T02:37:53.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2009-07-13T02:37:54.000-
04:00</DateFinishedReceivingTransmission>

```

#### 4.4.5.1.2 Integrated Partner Post Payload

##### 4.4.5.1.2.1 Post to Central Broker Payload

The “Post” payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED** xsd

#### **REDACTED**

```

  <!--Backfilled-->
<>EmployerTPABenefitChargesResponse><StateChargeRecordGUID>00000000000000000000
000000803</StateChargeRecordGUID><!--Backfilled-->
<>BrokerRecordTransactionNumber>6000</BrokerRecordTransactionNumber>
  <!--Backfilled--><NoticeDate>2016-11-08</NoticeDate>
  <!--Backfilled--><LastDayToRespond>2016-11-29</LastDayToRespond>
  <!--Backfilled--><ChargingStateAbbreviation>ST</ChargingStateAbbreviation>
  <!--Backfilled--><StateEmployerAccountNbr>8000001</StateEmployerAccountNbr>
  <!--Backfilled--><EmployerName>Third Test Company</EmployerName>
  <!--Backfilled--><FEIN>301416817</FEIN>
  <!--Backfilled--><ChargeStatementStartDate>2016-11-01</ChargeStatementStartDate>
  <!--Backfilled--><ChargeStatementEndDate>2016-11-09</ChargeStatementEndDate>
  <!--Required: 1-n times-->
  <SSNBYBChargeProgramCode>
    <!--Backfilled-->
    <SSN>123456789</SSN>
    <!--Backfilled-->
    <ClaimantLastName>Edwards</ClaimantLastName>

```



<!--Backfilled-->  
 <ClaimantFirstName>Jenna</ClaimantFirstName>  
 <!--Backfilled-->  
 <ClaimantMiddleInitial>N</ClaimantMiddleInitial>  
 <!--Backfilled-->  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this  
 company during this period-->  
 <!--11 = Claimant was working for some other employer during this period, 20 =  
 Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not  
 available for work, 40 = Claimant worked for employer less than the time period or earnings  
 required to be a chargeable employer--><!--41 = Charges are for a period of claimant  
 ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was  
 not received, 43 = Determination of eligibility not received-->  
 <!--44 = Employer is exempt from charges due to circumstances of separation, 46  
 = Credit has been previously identified and approved, but not received-->  
 <!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone  
 ever having worked for them under this name or SSN, 60 = Charges are being questioned due to  
 a pending protest/appeal of eligibility determination, 70 = Employer questions charges for  
 statutory reasons NOT listed above-->  
 <ProgramCode>1</ProgramCode>  
 <ProtestAppealQuestionReasonInd>10</ProtestAppealQuestionReasonInd>  
 <!--Required: Up to 2000 characters-->  
 <EmployerReasonForProtestAppealQuestion>claimant worked  
 here</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode><!--Required: 1-n times-->  
 <SSNBYBChargeProgramCode>  
 <!--Backfilled-->  
 <SSN>456932654</SSN>  
 <!--Backfilled-->  
 <ClaimantLastName>Johns</ClaimantLastName>  
 <!--Backfilled-->  
 <ClaimantFirstName>Jenna</ClaimantFirstName>  
 <!--Backfilled-->  
 <ClaimantMiddleInitial>K</ClaimantMiddleInitial>  
 <!--Backfilled-->  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <!--Backfilled-->  
 <ProgramCode>2</ProgramCode>  
 <!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this  
 company during this period-->  
 <!--11 = Claimant was working for some other employer during this period, 20 =  
 Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not  
 available for work, 40 = Claimant worked for employer less than the time period or earnings  
 required to be a chargeable employer--><!--41 = Charges are for a period of claimant  
 ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was  
 not received, 43 = Determination of eligibility not received-->  
 <!--44 = Employer is exempt from charges due to circumstances of separation, 46  
 = Credit has been previously identified and approved, but not received-->  
 <!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone  
 ever having worked for them under this name or SSN, 60 = Charges are being questioned due to

a pending protest/appeal of eligibility determination, 70 = Employer questions charges for statutory reasons NOT listed above-->

<ProtestAppealQuestionReasonInd>51</ProtestAppealQuestionReasonInd>

<!--Required: Up to 2000 characters-->

<EmployerReasonForProtestAppealQuestion>Jenna Johns has not worked here</EmployerReasonForProtestAppealQuestion>

</SSNBYBChargeProgramCode>

<!--Required: 1-n times-->

<SSNBYBChargeProgramCode>

<!--Backfilled-->

<SSN>203968375</SSN>

<!--Backfilled-->

<ClaimantLastName>Douglass</ClaimantLastName>

<!--Backfilled-->

<ClaimantFirstName>Frederick</ClaimantFirstName>

<!--Backfilled-->

<ClaimantMiddleInitial>A</ClaimantMiddleInitial>

<!--Backfilled-->

<BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>

<!--Backfilled-->

<ProgramCode>3</ProgramCode>

<!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this company during this period-->

<!--11 = Claimant was working for some other employer during this period, 20 = Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not available for work, 40 = Claimant worked for employer less than the time period or earnings required to be a chargeable employer--><!--41 = Charges are for a period of claimant ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was not received, 43 = Determination of eligibility not received-->

<!--44 = Employer is exempt from charges due to circumstances of separation, 46 = Credit has been previously identified and approved, but not received-->

<!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone ever having worked for them under this name or SSN, 60 = Charges are being questioned due to a pending protest/appeal of eligibility determination, 70 = Employer questions charges for statutory reasons NOT listed above-->

<ProtestAppealQuestionReasonInd>10</ProtestAppealQuestionReasonInd>

<!--Required: Up to 2000 characters-->

<EmployerReasonForProtestAppealQuestion>Frederick Douglass worked here</EmployerReasonForProtestAppealQuestion>

</SSNBYBChargeProgramCode>

<!--Required: 1-n times-->

<SSNBYBChargeProgramCode>

<!--Backfilled-->

<SSN>217262897</SSN>

<!--Backfilled-->

<ClaimantLastName>Kerry</ClaimantLastName>

<!--Backfilled-->

<ClaimantFirstName>Marcia</ClaimantFirstName>

<!--Backfilled-->

<ClaimantMiddleInitial>B</ClaimantMiddleInitial>

<!--Backfilled-->  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <!--Backfilled-->  
 <ProgramCode>4</ProgramCode>  
 <!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this  
 company during this period-->  
 <!--11 = Claimant was working for some other employer during this period, 20 =  
 Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not  
 available for work, 40 = Claimant worked for employer less than the time period or earnings  
 required to be a chargeable employer--><!--41 = Charges are for a period of claimant  
 ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was  
 not received, 43 = Determination of eligibility not received-->  
 <!--44 = Employer is exempt from charges due to circumstances of separation, 46  
 = Credit has been previously identified and approved, but not received-->  
 <!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone  
 ever having worked for them under this name or SSN, 60 = Charges are being questioned due to  
 a pending protest/appeal of eligibility determination, 70 = Employer questions charges for  
 statutory reasons NOT listed above-->  
 <ProtestAppealQuestionReasonInd>41</ProtestAppealQuestionReasonInd>  
 <!--Required: Up to 2000 characters-->  
 <EmployerReasonForProtestAppealQuestion>See previous  
 DD</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode>

<!--Required: 1-n times-->  
 <SSNBYBChargeProgramCode>  
 <!--Backfilled-->  
 <SSN>217259876</SSN>  
 <!--Backfilled-->  
 <ClaimantLastName>Jones</ClaimantLastName>  
 <!--Backfilled-->  
 <ClaimantFirstName>Michael</ClaimantFirstName>  
 <!--Backfilled-->  
 <ClaimantMiddleInitial>E</ClaimantMiddleInitial>  
 <!--Backfilled-->  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <ProgramCode>5</ProgramCode>  
 <!--Required for each entry: Up to 3 digits: 10 = Claimant was working for this  
 company during this period-->  
 <!--11 = Claimant was working for some other employer during this period, 20 =  
 Claimant refused to work for the employer when requested, 30 = Claimant is not able to or is not  
 available for work, 40 = Claimant worked for employer less than the time period or earnings  
 required to be a chargeable employer--><!--41 = Charges are for a period of claimant  
 ineligibility based upon a previously issued determination/decisions, 42 = A notice of claim was  
 not received, 43 = Determination of eligibility not received-->  
 <!--44 = Employer is exempt from charges due to circumstances of separation, 46  
 = Credit has been previously identified and approved, but not received-->  
 <!--50 = Wages reported are incorrect, 51 = Employer has no record of anyone  
 ever having worked for them under this name or SSN, 60 = Charges are being questioned due to  
 a pending protest/appeal of eligibility determination, 70 = Employer questions charges for  
 statutory reasons NOT listed above-->  
 <ProtestAppealQuestionReasonInd>10</ProtestAppealQuestionReasonInd>

```

        <!--Required: Up to 2000 characters-->
        <EmployerReasonForProtestAppealQuestion>claimant worked
here</EmployerReasonForProtestAppealQuestion>
        </SSNBYBChargeProgramCode>

        <!--Conditional: 1 digit: 1 = No, 2 = Agent, 3 = Attorney-->
        <AgentType>1</AgentType>
        <!--Required: 1 character: E = Employer, T = Third Party Administrator-->
        <PreparerTypeCode>E</PreparerTypeCode><!--Required: Up to 15 digits-->
        <PreparerTelephoneNumberPlusExt>7036634563</PreparerTelephoneNumberPlusExt>
        <!--Required: Up to 60 characters--><PreparerContactName>Jason
Smith</PreparerContactName>
        <!--Required: Up to 30 characters--><PreparerTitle>HR Manager</PreparerTitle>
        <!--Required: Up to 60 characters-->
><PreparerEmailAddress>jsmith@testco3.com</PreparerEmailAddress>
        </EmployerTPABenefitChargesResponse>

```

#### 4.4.5.1.2.2 Central Broker Acknowledgement to Integrated Partner Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of responses it received the number in error, and the dates that it started receiving the records and finished receiving the records. This verifies to the Integrated Partner that the Broker received the desired file so it can move on to the next file.

#### REDACTED

```

        <EmployerTPAResponseFileGUID>542A4AE2395FEDDF1EAA3E57F2DFBCE0</EmployerTPARespo
nseFileGUID>
        <NumberOfResponseRecordsReceived>22</NumberOfResponseRecordsReceived>
        <NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
        <DateStartedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateStartedReceivingTransmission>
        <DateFinishedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateFinishedReceivingTransmission>
</EmployerTPABenefitChargesResponseCollectionAcknowledgement>

```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the “Post” message, the Broker will also return to the Integrated Partner all of the information that it can on why each individual response failed. The BenefitChargesResponse element defined in the Benefit Charges Response XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```

<FailedBenefitChargesResponse>
        <StateBenefitChargesRequestRecordGUID>0000000000000000000000000099943</StateBen

```

```

efitChargesRequestRecordGUID>
  <BrokerRecordTransactionNumber>2001552</BrokerRecordTransactionNumber>
  <ErrorOccurrence>
    <ErrorCode>1013</ErrorCode>
    <ErrorMessage> Business Rule violation - the AgentType (Character - size 1) is
required if request AgentResponseIndicator = 1.</ErrorMessage>
  </ErrorOccurrence>
</FailedMonetaryChargeResponse>

```

Putting this together with the successful acknowledgement:

### REDACTED

```

<EmployerTPAResponseFileGUID>542A4AE2395FEDDF1EAA3E57F2DFBCE0</EmployerTPARespo
nseFileGUID>

<FailedBenefitChargesResponse>
  <StateBenefitChargesRequestRecordGUID>0000000000000000000000000099943</StateBen
efitChargesRequestRecordGUID>
  <BrokerRecordTransactionNumber>2001552</BrokerRecordTransactionNumber>
  <ErrorOccurrence>
    <ErrorCode>1013</ErrorCode>
    <ErrorMessage> Business Rule violation - the AgentType (Character - size
1) is required if request AgentResponseIndicator = 1.</ErrorMessage>
  </ErrorOccurrence>
</FailedBenefitChargesResponse>
  <NumberOfResponseRecordsReceived>22</NumberOfResponseRecordsReceived>
  <NumberOfResponseRecordsInError>1</NumberOfResponseRecordsInError>
  <DateStartedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2009-07-22T03:13:50.000-
04:00</DateFinishedReceivingTransmission>

```

#### 4.4.5.2 Pull Payload

A “Pull” is defined (see Section 4.1.2) as sending a query to the Central Broker to allow the Central Broker to send the Connector all of the requests or responses it has waiting for it (in multiple transactions if they are from different endpoints or if the data exceeds 8MB). This section discusses the (pre-encryption) payloads in the SOAP message.

##### 4.4.5.2.1 State Pull Payload

###### 4.4.5.2.1.1 Pull from Central Broker

The **REDACTED** defined in the State Benefit Charges WSDL (see Section 4.6) is a complex type that allows the caller to specify one of three operations: a Pull, a Re-Pull by StateSOAPTransactionNumber, and a Re-Pull by a Date Range.

```
<!-- Query element for states to collect claim responses they are expecting -->
```

### REDACTED

For the regular Pull, the caller needs to supply only the State Unique ID. Although there are different ways to verify the calling State besides this element, the Broker uses it as an additional security check. Also, there is a requirement in the WSDL 1.1 specification that a WSDL definition have at least one input attribute.

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
```

For the Re-Pull by StateSOAPTransactionNumber, the caller needs to supply the State Unique ID and the StateSOAPTransactionNumber element out of the **REDACTED**. This will allow the Broker to send the file defined by the StateSOAPTransactionNumber.

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
<StateBenefitChargesResponseCollectionQueryCriteria>  
  <StateSOAPTransactionNumber>123456221</StateSOAPTransactionNumber>  
</StateBenefitChargesResponseCollectionQueryCriteria>
```

For the Re-Pull by date range, the caller needs to supply the State Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and a StateSOAPTransactionNumber.

The first time this operation is called, the StateSOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next StateSOAPTransactionNumber that it sent during that date range (in element name

NextStateSOAPTransactionNumber). In the next call to this operation, the caller must include the NextStateSOAPTransactionNumber as the StateSOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next StateSOAPTransactionNumber (so there will not be a NextStateSOAPTransactionNumber included in the http response SOAP header).

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

First Call:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
<StateBenefitChargesResponseCollectionQueryCriteria>  
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-  
04:00</BrokerRecordEffectiveDateFrom>  
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>  
</StateBenefitChargesResponseCollectionQueryCriteria>
```

Subsequent Calls:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
<StateBenefitChargesResponseCollectionQueryCriteria>  
  <StateSOAPTransactionNumber>12345678901234567890123456789012</StateSOAPTransa  
ctionNumber>  
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-  
04:00</BrokerRecordEffectiveDateFrom>  
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-  
04:00</BrokerRecordEffectiveDateTo>  
</StateBenefitChargesResponseCollectionQueryCriteria>
```

#### 4.4.5.2.1.2 Central Broker Response to the State

When the Broker receives a “Pull” request from a State, it begins assembling all the responses that are intended for that State. It constructs a SOAP message according to the rules for a Benefit Charges Information SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds the additional field BrokerRecordEffectiveDate to the response, which indicates the date that the response was accepted into the Broker. It then sends the responses in the HTTP response.

**REDACTED**

```
<StateBenefitChargesResponse>  
  
<StateChargeRecordGUID>00000000000000000000000000000803</StateChargeRecordGUID>  
  <BrokerRecordTransactionNumber>6000</BrokerRecordTransactionNumber>  
  <NoticeDate>2016-11-08</NoticeDate>  
  <LastDayToRespond>2016-11-29</LastDayToRespond>  
  <ChargingStateAbbreviation>ST</ChargingStateAbbreviation>  
  <StateEmployerAccountNbr>8000001</StateEmployerAccountNbr>  
  <EmployerName>Third Test Company</EmployerName>  
  <FEIN>301416817</FEIN>  
  <ChargeStatementStartDate>2016-11-01</ChargeStatementStartDate>  
  <ChargeStatementEndDate>2016-11-09</ChargeStatementEndDate>  
  <SSNBYBChargeProgramCode>  
    <SSN>123456789</SSN>  
    <ClaimantLastName>Edwards</ClaimantLastName>  
    <ClaimantFirstName>Jenna</ClaimantFirstName>
```

<ClaimantMiddleInitial>N</ClaimantMiddleInitial>  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <ProgramCode>1</ProgramCode>  
 <ProtestAppealQuestionReasonInd>10</ProtestAppealQuestionReasonInd>  
 <EmployerReasonForProtestAppealQuestion>claimant worked  
 here</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode>  
 <SSNBYBChargeProgramCode>  
 <SSN>456932654</SSN>  
 <ClaimantLastName>Johns</ClaimantLastName>  
 <ClaimantFirstName>Jenna</ClaimantFirstName>  
 <ClaimantMiddleInitial>K</ClaimantMiddleInitial>  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <ProgramCode>2</ProgramCode>  
 <ProtestAppealQuestionReasonInd>51</ProtestAppealQuestionReasonInd>  
 <EmployerReasonForProtestAppealQuestion>Jenna Johns has not worked  
 here</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode>  
 <SSNBYBChargeProgramCode>  
 <SSN>203968375</SSN>  
 <ClaimantLastName>Douglass</ClaimantLastName>  
 <ClaimantFirstName>Frederick</ClaimantFirstName>  
 <ClaimantMiddleInitial>A</ClaimantMiddleInitial>  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <ProgramCode>3</ProgramCode>  
 <ProtestAppealQuestionReasonInd>10</ProtestAppealQuestionReasonInd>  
 <EmployerReasonForProtestAppealQuestion>Frederick Douglass worked  
 here</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode>  
 <SSNBYBChargeProgramCode>  
 <SSN>217262897</SSN>  
 <ClaimantLastName>Kerry</ClaimantLastName>  
 <ClaimantFirstName>Marcia</ClaimantFirstName>  
 <ClaimantMiddleInitial>B</ClaimantMiddleInitial>  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <ProgramCode>4</ProgramCode>  
 <ProtestAppealQuestionReasonInd>41</ProtestAppealQuestionReasonInd>  
 <EmployerReasonForProtestAppealQuestion>See previous  
 DD</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode>  
 <SSNBYBChargeProgramCode>  
 <SSN>217259876</SSN>  
 <ClaimantLastName>Jones</ClaimantLastName>  
 <ClaimantFirstName>Michael</ClaimantFirstName>  
 <ClaimantMiddleInitial>E</ClaimantMiddleInitial>  
 <BenefitYearBeginDate>2016-10-24</BenefitYearBeginDate>  
 <ProgramCode>5</ProgramCode>  
 <ProtestAppealQuestionReasonInd>10</ProtestAppealQuestionReasonInd>  
 <EmployerReasonForProtestAppealQuestion>claimant worked  
 here</EmployerReasonForProtestAppealQuestion>  
 </SSNBYBChargeProgramCode>  
 <AgentType>1</AgentType>



```
<PreparerTypeCode>E</PreparerTypeCode>
<PreparerTelephoneNumberPlusExt>7036634563</PreparerTelephoneNumberPlusExt>
<PreparerContactName>Jason Smith</PreparerContactName>
<PreparerTitle>HR Manager</PreparerTitle>
<PreparerEmailAddress>jsmith@testco3.com</PreparerEmailAddress>
<BrokerRecordEffectiveDate>2016-09-02T07:40:33.000-04:00</BrokerRecordEffectiveDate>
</StateBenefitChargesResponse>
```

#### 4.4.5.2.1.3 State Acknowledgement to the Central Broker

The **REDACTED** is initiated once the State has received its file from the Broker. The acknowledgement must accompany every State Pull request, as this is the manner in which the Broker knows that the State Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next "Pull" call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the State until the NASWA SIDES team and State Administrator can work out the problem.

The key field in this message is the StateSOAPTransmissionNumber, which must correspond with the StateSOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the State does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

#### **REDACTED**

```
<StateSOAPTransmissionNumber>12345678901234567890123456789012</StateSOAPTransmissionNumber>
<NumberOfResponseRecordsReceived>0</NumberOfResponseRecordsReceived>
<NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
<DateStartedReceivingTransmission>2008-12-31T12:00:00.000-04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2008-12-31T12:00:00.000-04:00</DateFinishedReceivingTransmission>
```

#### 4.4.5.2.2 Integrated Partner Pull

##### 4.4.5.2.2.1 Pull from Central Broker

The **REDACTED** is a complex query type that allows the caller to specify one of three operations: a Pull, a Re-Pull by EmployerTPASOAPTransactionNumber, and a Re-Pull by a Date Range.

<!-- Query element for employer to collect claim responses they are expecting -->

**REDACTED**

For the regular Pull, the caller needs to supply only the Integrated Partner Unique ID. Although there are different ways to verify the calling Integrated Partner besides this element, the Broker uses it as an additional security check. Also, there is a requirement in WSDL 1.1 that a WSDL definition have at least one input attribute.

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>
```

For the Re-Pull by EmployerTPASOAPTransactionNumber, the caller needs to supply the Integrated Partner Unique ID and the EmployerTPASOAPTransactionNumber element out of the **REDACTED**. This will allow the Broker to send the file defined by the EmployerTPASOAPTransactionNumber.

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPABenefitChargesRequestCollectionQueryCriteria>
  <EmployerTPASOAPTransactionNumber>26151</EmployerTPASOAPTransactionNumber>
</EmployerTPABenefitChargesRequestCollectionQueryCriteria>
```

For the Re-Pull by date range, the caller needs to supply the Integrated Partner Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex query type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and an EmployerTPASOAPTransactionNumber.

The first time this operation is called, the EmployerTPASOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next EmployerTPASOAPTransactionNumber that it sent during that date range (in element name NextEmployerTPASOAPTransactionNumber). In the next call to this operation, the caller must include the NextEmployerTPASOAPTransactionNumber as the EmployerTPASOAPTransactionNumber along with the date range. This

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next EmployerTPASOAPTransactionNumber (so there will not be a NextEmployerTPASOAPTransactionNumber included in the http response SOAP header).

First Call:

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPABenefitChargesRequestCollectionQueryCriteria>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>
</EmployerTPABenefitChargesRequestCollectionQueryCriteria>
```

Subsequent Calls:

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPABenefitChargesRequestCollectionQueryCriteria>
  <EmployerTPASOAPTransactionNumber>12345678901234567890123456789012</Employer
TPASOAPTransactionNumber>
  <BrokerRecordEffectiveDateFrom>2009-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2009-12-31T12:00:00.000-
04:00</BrokerRecordEffectiveDateTo>
</EmployerTPABenefitChargesRequestCollectionQueryCriteria>
```

#### 4.4.5.2.2 Central Broker Response to Integrated Partner

When the Broker receives a "Pull" request from an Integrated Partner, it begins assembling all the requests that are intended for that Integrated Partner. It constructs a SOAP message according to the rules for a Benefit Charges Information SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds two additional fields to the Benefit Charges Request - the BrokerRecordEffectiveDate and the BrokerRecordTransactionNumber. The BrokerRecordEffectiveDate indicates the date that the request was accepted into the Broker. The BrokerRecordTransactionNumber is a unique record tracking number and must be returned on the response for this record. It then sends the Benefit Charges requests in the HTTP response.

**REDACTED**

```
<EmployerTPABenefitChargesRequest>
```

<StateChargeRecordGUID>8000000000000000000000000000942</StateChargeRecordGUID>  
<NoticeDate>2016-09-01</NoticeDate>  
<DueDateRequiredInd>N</DueDateRequiredInd>  
<ChargingStateAbbreviation>ST</ChargingStateAbbreviation>  
<UIOfficeName>Test UI Office 942</UIOfficeName>  
<ExperienceRatingMethod>1</ExperienceRatingMethod>  
<StateEmployerAccountNbr>8000000000942</StateEmployerAccountNbr>  
<EmployerName>Employer 942</EmployerName>  
<FEIN>462904950</FEIN>  
<ChargeStatementStartDate>2016-08-21</ChargeStatementStartDate>  
<ChargeStatementEndDate>2016-10-21</ChargeStatementEndDate>  
<EmployerAccountType>1</EmployerAccountType>  
<NumberOfIndividualSSNsWithCharges>1</NumberOfIndividualSSNsWithCharges>  
<TotalDollarAmountChargedToEmployer>200.00</TotalDollarAmountChargedToEmployer>  
<ReserveBalance>5000.00</ReserveBalance>  
<SSNBYBChargeProgramCode>  
<SSN>217270942</SSN>  
<BenefitYearBeginDate>2016-07-14</BenefitYearBeginDate>  
<ProgramCode>1</ProgramCode>  
<ClaimantLastName>Jones</ClaimantLastName>  
<ClaimantFirstName>Fred</ClaimantFirstName>  
<ClaimantMiddleInitial>K</ClaimantMiddleInitial>  
<BasePeriodStartDate>2015-07-07</BasePeriodStartDate>  
<BasePeriodEndDate>2016-07-12</BasePeriodEndDate>  
<TotalAmountChargedToEmployer>200.00</TotalAmountChargedToEmployer>

<TotalNumberOfWeeksChargedToEmployerAccount>1</TotalNumberOfWeeksChargedToEmployerAccount>  
</SSNBYBChargeProgramCode>  
<AgentResponseIndicator>2</AgentResponseIndicator>  
<NotAvailableIndicator>2</NotAvailableIndicator>  
<InterpreterIndicator>2</InterpreterIndicator>  
<SpecialNeedsIndicator>2</SpecialNeedsIndicator>  
<StateTerminologyUsed>2</StateTerminologyUsed>  
<BrokerRecordTransactionNumber>9001</BrokerRecordTransactionNumber>  
<BrokerRecordEffectiveDate>2016-08-30T19:42:07.000-04:00</BrokerRecordEffectiveDate>  
</EmployerTPABenefitChargesRequest>  
<EmployerTPABenefitChargesRequest>

<StateChargeRecordGUID>8000000000000000000000000000927</StateChargeRecordGUID>  
<NoticeDate>2016-09-01</NoticeDate>  
<DueDateRequiredInd>N</DueDateRequiredInd>  
<ChargingStateAbbreviation>ST</ChargingStateAbbreviation>  
<UIOfficeName>Test UI Office 927</UIOfficeName>  
<ExperienceRatingMethod>1</ExperienceRatingMethod>  
<StateEmployerAccountNbr>8000000000927</StateEmployerAccountNbr>  
<EmployerName>Employer 927</EmployerName>  
<FEIN>462904950</FEIN>  
<ChargeStatementStartDate>2016-08-21</ChargeStatementStartDate>  
<ChargeStatementEndDate>2016-10-21</ChargeStatementEndDate>  
<EmployerAccountType>1</EmployerAccountType>

```

<NumberOfIndividualSSNsWithCharges>1</NumberOfIndividualSSNsWithCharges>
<TotalDollarAmountChargedToEmployer>200.00</TotalDollarAmountChargedToEmployer>
<ReserveBalance>5000.00</ReserveBalance>
<SSNBYBChargeProgramCode>
  <SSN>217270927</SSN>
  <BenefitYearBeginDate>2016-08-02</BenefitYearBeginDate>
  <ClaimantLastName>Jones</ClaimantLastName>
  <ClaimantFirstName>Fred</ClaimantFirstName>
  <ClaimantMiddleInitial>K</ClaimantMiddleInitial>
  <BasePeriodStartDate>2015-07-07</BasePeriodStartDate>
  <BasePeriodEndDate>2016-07-12</BasePeriodEndDate>
  <TotalAmountChargedToEmployer>200.00</TotalAmountChargedToEmployer>

<TotalNumberOfWeeksChargedToEmployerAccount>1</TotalNumberOfWeeksChargedToEmployerAccount>
  </SSNBYBChargeProgramCode>
  <AgentResponseIndicator>2</AgentResponseIndicator>
  <NotAvailableIndicator>2</NotAvailableIndicator>
  <InterpreterIndicator>2</InterpreterIndicator>
  <SpecialNeedsIndicator>2</SpecialNeedsIndicator>
  <StateTerminologyUsed>2</StateTerminologyUsed>
  <BrokerRecordTransactionNumber>9000</BrokerRecordTransactionNumber>
  <BrokerRecordEffectiveDate>2016-08-30T19:00:23.000-04:00</BrokerRecordEffectiveDate>
</EmployerTPABenefitChargesRequest>

```

#### 4.4.5.2.3 Integrated Partner Acknowledgement to Central Broker

The **REDACTED** is initiated once the Integrated Partner has received its file from the Broker. The acknowledgement must accompany every Integrated Partner Pull request, as this is the manner in which the Broker knows that the Integrated Partner Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next "Pull" call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the Integrated Partner until the NASWA SIDES team and Integrated Partner Administrator can work out the problem.

The key field in this message is the EmployerTPASOAPTransmissionNumber, which must correspond with the EmployerTPASOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the Integrated Partner does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

#### **REDACTED**

```

<EmployerTPASOAPTransmissionNumber>3211</EmployerTPASOAPTransmissionNumber>
<NumberOfRequestRecordsReceived>4</NumberOfRequestRecordsReceived>

```

```
<NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2001-12-31T12:00:00.000-
04:00</DateFinishedReceivingTransmission>
```

## 4.4.6 Additional Fact-Finding

### 4.4.6.1 Post Payload

A "Post" is defined (see Section 4.1.1) as sending a request or response to the Broker by a particular Connector. This section discusses the (pre-encryption) payloads in the SOAP message.

#### 4.4.6.1.1 State Post Payload

##### 4.4.6.1.1.1 Post to Central Broker Payload

The "Post" payload in the SOAP message is the data defined in the AdditionalFactFindingRequestCollection defined in the AdditionalFactFindingRequest xsd.

### REDACTED

```
<EmployerTPAAdditionalFactFindingResponse>
  <StateRequestRecordGUID>11000000000000000000000000000001</StateRequestRecordGUID>
  <BrokerRecordTransactionNumber>13156</BrokerRecordTransactionNumber>
  <RequestDate>2019-09-10</RequestDate>
  <ResponseDueDate>2019-09-21</ResponseDueDate>
  <SSN>000110001</SSN>
  <ClaimEffectiveDate>2019-09-10</ClaimEffectiveDate>
  <ClaimNumber>11231</ClaimNumber>
  <StateEmployerAccountNbr>179424876</StateEmployerAccountNbr>
  <EmployerName>Pennies Inc</EmployerName>
  <ClaimantLastName>Smith</ClaimantLastName>
  <ClaimantFirstName>Ron</ClaimantFirstName>
  <ClaimantMiddleInitial>S</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>
  <AdjudicatorIdentifier>ADJ101</AdjudicatorIdentifier>
  <RequestType>1</RequestType>
  <NumberOfInquiries>3</NumberOfInquiries>
  <InquiryOccurrence>
    <InquiryNumber>01</InquiryNumber>
    <InquiryQuestion>You indicated the claimant was discharge for drugs. What made you
believe the claimant was on drugs</InquiryQuestion>
    <ResponseComments>Claimant Doctor Reports</ResponseComments>
  </InquiryOccurrence>
  <InquiryOccurrence>
    <InquiryNumber>02</InquiryNumber>
    <InquiryQuestion>Do you have a drug testing policy. If yes provide policy</InquiryQuestion>
    <ResponseComments>Yes, Policy handbook attached sent via email on
request</ResponseComments>
  </InquiryOccurrence>
</InquiryOccurrence>
```



```

    <ErrorCode>1111</ErrorCode>
    <ErrorMessage>Business Rule violation - RequestDate (date) must be the day after the due
date of the original response or immediately following the receipt the employer
response.</ErrorMessage>
  </ErrorOccurrence>
  <ErrorOccurrence>
    <ErrorCode>1118</ErrorCode>
    <ErrorMessage>Business Rule violation - OtherExchangeBRTN (character - size 32) due date is
in the future and no response record has been found.</ErrorMessage>
  </ErrorOccurrence>

```

Putting this together with the successful acknowledgement:

### REDACTED

```

  <StateRequestFileGUID
xmlns="https://uidataexchange.org/schemas">12345678901234567890123456780000</StateRequ
estFileGUID>
  <FailedAdditionalFactFindingRequest xmlns="https://uidataexchange.org/schemas">

<StateAdditionalFactFindingRequestRecordGUID>110000000000000000000000000000001111</StateA
dditionalFactFindingRequestRecordGUID>
  <ErrorOccurrence>
    <ErrorCode>1111</ErrorCode>
    <ErrorMessage>Business Rule violation - RequestDate (date) must be the day after the due
date of the original response or immediately following the receipt the employer
response.</ErrorMessage>
  </ErrorOccurrence>
  <ErrorOccurrence>
    <ErrorCode>1118</ErrorCode>
    <ErrorMessage>Business Rule violation - OtherExchangeBRTN (character - size 32) due date is
in the future and no response record has been found.</ErrorMessage>
  </ErrorOccurrence>
</FailedAdditionalFactFindingRequest>
  <NumberOfRequestRecordsReceived
xmlns="https://uidataexchange.org/schemas">1</NumberOfRequestRecordsReceived>
  <NumberOfRequestRecordsInError
xmlns="https://uidataexchange.org/schemas">1</NumberOfRequestRecordsInError>
  <DateStartedReceivingTransmission xmlns="https://uidataexchange.org/schemas">2019-08-
30T06:15:18.594-04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission xmlns="https://uidataexchange.org/schemas">2019-08-
30T06:15:18.649-04:00</DateFinishedReceivingTransmission>

```

#### 4.4.6.1.2 Integrated Partner Post Payload

##### 4.4.6.1.2.1 Post to Central Broker Payload

The “Post” payload in the SOAP message is the data defined in the **REDACTED** defined in the **REDACTED** xsd.

### REDACTED





```

<EmployerTPAAdditionalFactFindingResponse>
  <StateRequestRecordGUID>11000000000000000000000000000001</StateRequestRecordGUID>
  <BrokerRecordTransactionNumber>13156</BrokerRecordTransactionNumber>
  <RequestDate>2019-09-10</RequestDate>
  <ResponseDueDate>2019-09-21</ResponseDueDate>
  <SSN>000110001</SSN>
  <ClaimEffectiveDate>2019-09-10</ClaimEffectiveDate>
  <ClaimNumber>11231</ClaimNumber>
  <StateEmployerAccountNbr>179424876</StateEmployerAccountNbr>
  <EmployerName>Pennies Inc</EmployerName>
  <ClaimantLastName>Smith</ClaimantLastName>
  <ClaimantFirstName>Ron</ClaimantFirstName>
  <ClaimantMiddleInitial>S</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>
  <AdjudicatorIdentifier>ADJ101</AdjudicatorIdentifier>
  <RequestType>1</RequestType>
  <NumberOfInquiries>3</NumberOfInquiries>
  <InquiryOccurrence>
    <InquiryNumber>01</InquiryNumber>
    <InquiryQuestion>You indicated the claimant was discharge for drugs. What made you
believe the claimant was on drugs</InquiryQuestion>
    <ResponseComments>Claimant Doctor Reports</ResponseComments>
  </InquiryOccurrence>
  <InquiryOccurrence>
    <InquiryNumber>02</InquiryNumber>
    <InquiryQuestion>Do you have a drug testing policy. If yes provide policy</InquiryQuestion>
    <ResponseComments>Yes, Policy handbook attached sent via email on
request</ResponseComments>
  </InquiryOccurrence>
  <InquiryOccurrence>
    <InquiryNumber>03</InquiryNumber>
    <InquiryQuestion>Was the claimant given a drug test. If no why was the claimant not
tested.If yes what was the result of the drug test</InquiryQuestion>
    <ResponseComments>Yes, claimant was tested before hired</ResponseComments>
  </InquiryOccurrence>

  <PreparerTypeCode>E</PreparerTypeCode>
  <PreparerTelephoneNumberPlusExt>33333333333333</PreparerTelephoneNumberPlusExt>
  <PreparerContactName>Ronny Melford</PreparerContactName>
  <PreparerTitle>Preparor Level1</PreparerTitle>
  <PreparerEmailAddress>Ronny@asdf.com</PreparerEmailAddress>
</EmployerTPAAdditionalFactFindingResponse>

```

#### 4.4.6.1.2.2 Central Broker Acknowledgement to Integrated Partner Payload

In the acknowledgement to the State Post, the Broker sends back a response that contains the number of responses it received, the number in error, and the dates that it started receiving the records and finished receiving the records. This verifies to the Integrated Partner that the Broker received the desired file so it can move on to the next file.

**REDACTED**

```
<EmployerTPAResponseFileGUID>12345678901234567890123456781000</EmployerTPAResponseFileGUID>  
<NumberOfResponseRecordsReceived>1</NumberOfResponseRecordsReceived>  
<NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>  
<DateStartedReceivingTransmission>2019-09-04T12:50:12.771-04:00</DateStartedReceivingTransmission>  
<DateFinishedReceivingTransmission>2019-09-04T12:50:13.012-04:00</DateFinishedReceivingTransmission>
```

**Note: The Broker also sends back custom SOAP header information that tells the overall status of the message. This is defined in Section 4.3 SOAP Custom Headers.**

If the Broker determines there were Business Rule or XSD errors in the "Post" message, the Broker will also return to the Integrated Partner all of the information that it can on why each individual response failed. The FailedAdditionalFactFindingResponse element defined in the Additional Fact-Finding response XSD will present the Error Code and the Error Message of the error it found, as described in Part B, Section C-2.8.

```
<FailedAdditionalFactFindingResponse>  
  <StateAdditionalFactFindingRequestRecordGUID>11000000000000000000000000001210</StateAdditionalFactFindingRequestRecordGUID>  
  <BrokerRecordTransactionNumber>13156</BrokerRecordTransactionNumber>  
  <ErrorOccurrence>  
    <ErrorCode>1210</ErrorCode>  
    <ErrorMessage>Business Rule violation - There is no matching Claim Request record with fields matching the backfilled fields, the StateRequestRecordGUID, and the BrokerRecordTransactionNumber.</ErrorMessage>  
  </ErrorOccurrence>  
  <ErrorOccurrence>  
    <ErrorCode>1214</ErrorCode>  
    <ErrorMessage>Responding Employer/TPA does not match the Employer/TPA to which the request was sent.</ErrorMessage>  
  </ErrorOccurrence>  
  <ErrorOccurrence>  
    <ErrorCode>1215</ErrorCode>  
    <ErrorMessage>State to which the Employer/TPA is responding does not match the State from which the request was sent.</ErrorMessage>  
  </ErrorOccurrence>  
</FailedAdditionalFactFindingResponse>
```

Putting this together with the successful acknowledgement:

**REDACTED**

```

<EmployerTPAResponseFileGUID>12345678901234567890123456781000</EmployerTPAResponseFileGUID>
  <FailedAdditionalFactFindingResponse>
    <StateAdditionalFactFindingRequestRecordGUID>110000000000000000000000000000001210</StateAdditionalFactFindingRequestRecordGUID>
    <BrokerRecordTransactionNumber>13156</BrokerRecordTransactionNumber>
    <ErrorOccurrence>
      <ErrorCode>1210</ErrorCode>
      <ErrorMessage>Business Rule violation - There is no matching Claim Request record with fields matching the backfilled fields, the StateRequestRecordGUID, and the BrokerRecordTransactionNumber.</ErrorMessage>
    </ErrorOccurrence>
    <ErrorOccurrence>
      <ErrorCode>1214</ErrorCode>
      <ErrorMessage>Responding Employer/TPA does not match the Employer/TPA to which the request was sent.</ErrorMessage>
    </ErrorOccurrence>
    <ErrorOccurrence>
      <ErrorCode>1215</ErrorCode>
      <ErrorMessage>State to which the Employer/TPA is responding does not match the State from which the request was sent.</ErrorMessage>
    </ErrorOccurrence>
  </FailedAdditionalFactFindingResponse>
<NumberOfResponseRecordsReceived>1</NumberOfResponseRecordsReceived>
<NumberOfResponseRecordsInError>1</NumberOfResponseRecordsInError>
<DateStartedReceivingTransmission>2019-09-04T12:50:12.771-04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2019-09-04T12:50:13.012-04:00</DateFinishedReceivingTransmission>

```

#### 4.4.6.2 Pull Payload

A “Pull” is defined (see Section 4.1.2) as sending a query to the Central Broker to allow the Central Broker to send the Connector all of the requests or responses it has waiting for it (in multiple transactions if they are from different endpoints or if the data exceeds 8MB). This section discusses the (pre-encryption) payloads in the SOAP message.

##### 4.4.6.2.1 State Pull Payload

###### 4.4.6.2.1.1 Pull from Central Broker

The **REDACTED** defined in the State Additional Fact-Finding WSDL (see Section 4.6) is a complex type that allows the caller to specify one of three operations: a Pull, a Re-Pull by StateSOAPTransactionNumber, and a Re-Pull by a Date Range.

```

<!-- Query element for states to collect claim responses they are expecting -->
REDACTED

```

For the regular Pull, the caller needs to supply only the State Unique ID. Although there are different ways to verify the calling State besides this element, the Broker uses it as an additional security check. Also, there is a requirement in the WSDL 1.1 specification that a WSDL definition have at least one input attribute.

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
```

For the Re-Pull by StateSOAPTransactionNumber, the caller needs to supply the State Unique ID and the StateSOAPTransactionNumber element out of the **REDACTED**. This will allow the Broker to send the file defined by the StateSOAPTransactionNumber.

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>  
<StateAdditionalFactFindingResponseCollectionQueryCriteria>  
  <StateSOAPTransactionNumber>123456221</StateSOAPTransactionNumber>  
</StateAdditionalFactFindingResponseCollectionQueryCriteria>
```

For the Re-Pull by date range, the caller needs to supply the State Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and a StateSOAPTransactionNumber.

The first time this operation is called, the StateSOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next StateSOAPTransactionNumber that it sent during that date range (in element name NextStateSOAPTransactionNumber). In the next call to this operation, the caller must include the NextStateSOAPTransactionNumber as the StateSOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

StateSOAPTransactionNumber (so there will not be a NextStateSOAPTransactionNumber included in the http response SOAP header).

First Call:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
<StateAdditionalFactFindingResponseCollectionQueryCriteria>
  <BrokerRecordEffectiveDateFrom>2019-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2019-08-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>
</StateAdditionalFactFindingResponseCollectionQueryCriteria>
```

Subsequent Calls:

**REDACTED**

```
<StatePostalCode>ST</StatePostalCode>
<StateAdditionalFactFindingResponseCollectionQueryCriteria>
  <StateSOAPTransactionNumber>12345678901234567890123456789012</StateSOAPTransa
ctionNumber>
  <BrokerRecordEffectiveDateFrom>2019-01-01T12:00:00.000-
04:00</BrokerRecordEffectiveDateFrom>
  <BrokerRecordEffectiveDateTo>2019-08-31T12:00:00.000-
04:00</BrokerRecordEffectiveDateTo>
</StateAdditionalFactFindingResponseCollectionQueryCriteria>
```

#### 4.4.6.2.1.2 Central Broker Response to the State

When the Broker receives a “Pull” request from a State, it begins assembling all the responses that are intended for that State. It constructs a SOAP message according to the rules for an Additional Fact-Finding SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds the additional field BrokerRecordEffectiveDate to the response, which indicates the date that the response was accepted into the Broker. It then sends the responses in the HTTP response.

**REDACTED**

```
<ns2:StateAdditionalFactFindingResponse>
<ns2:StateRequestRecordGUID>11000000000000000000000000000001</ns2:StateRequestRecord
GUID>
  <ns2:BrokerRecordTransactionNumber>13003</ns2:BrokerRecordTransactionNumber>
  <ns2:RequestDate>2019-09-10</ns2:RequestDate>
  <ns2:ResponseDueDate>2019-09-21</ns2:ResponseDueDate>
  <ns2:SSN>000110001</ns2:SSN>
  <ns2:ClaimEffectiveDate>2019-09-10</ns2:ClaimEffectiveDate>
  <ns2:ClaimNumber>1231</ns2:ClaimNumber>
  <ns2:StateEmployerAccountNbr>179424876</ns2:StateEmployerAccountNbr>
  <ns2:EmployerName>Pennies Inc</ns2:EmployerName>
```

```

<ns2:ClaimantLastName>Ron</ns2:ClaimantLastName>
<ns2:ClaimantFirstName>Smith</ns2:ClaimantFirstName>
<ns2:ClaimantMiddleInitial>S</ns2:ClaimantMiddleInitial>
<ns2:ClaimantSuffix>Jr</ns2:ClaimantSuffix>
<ns2:AdjudicatorIdentifier>ADJ109</ns2:AdjudicatorIdentifier>
<ns2:RequestType>2</ns2:RequestType>
<ns2:NumberOfInquiries>2</ns2:NumberOfInquiries>
<ns2:InquiryOccurrence>
  <ns2:InquiryNumber>01</ns2:InquiryNumber>
  <ns2:InquiryQuestion>Why did you not work all available hours?</ns2:InquiryQuestion>
  <ns2:ResponseComments>asd</ns2:ResponseComments>
</ns2:InquiryOccurrence>
<ns2:InquiryOccurrence>
  <ns2:InquiryNumber>02</ns2:InquiryNumber>
  <ns2:InquiryQuestion>Who was your manager?</ns2:InquiryQuestion>
  <ns2:ResponseComments>asd</ns2:ResponseComments>
</ns2:InquiryOccurrence>
<ns2:PreparerTypeCode>E</ns2:PreparerTypeCode>

<ns2:PreparerTelephoneNumberPlusExt>3333333333333</ns2:PreparerTelephoneNumberPlusExt>
<ns2:PreparerContactName>asdf</ns2:PreparerContactName>
<ns2:PreparerTitle>asdf</ns2:PreparerTitle>
<ns2:PreparerEmailAddress>asdf@asdf.com</ns2:PreparerEmailAddress>
<ns2:BrokerRecordEffectiveDate>2019-08-23T17:51:23.000-
04:00</ns2:BrokerRecordEffectiveDate>
</ns2:StateAdditionalFactFindingResponse>

```

#### 4.4.6.2.1.3 State Acknowledgement to the Central Broker

The **REDACTED** is initiated once the State has received its file from the Broker. The acknowledgement must accompany every State Pull request, as this is the manner in which the Broker knows that the State Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next “Pull” call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the State until the NASWA SIDES team and State Administrator can work out the problem.

The key field in this message is the StateSOAPTransmissionNumber, which must correspond with the StateSOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the State does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

## REDACTED

```
<StateSOAPTransmissionNumber>12345678901234567890123456789012</StateSOAPTransmissionNumber>
  <NumberOfResponseRecordsReceived>0</NumberOfResponseRecordsReceived>
  <NumberOfResponseRecordsInError>0</NumberOfResponseRecordsInError>
  <DateStartedReceivingTransmission>2019-08-31T12:00:00.000-04:00</DateStartedReceivingTransmission>
  <DateFinishedReceivingTransmission>2019-08-31T12:00:00.000-04:00</DateFinishedReceivingTransmission>
```

### 4.4.6.2.2 Integrated Partner Pull

#### 4.4.6.2.2.1 Pull from Central Broker

The **REDACTED** is a complex query type that allows the caller to specify one of three operations: a Pull, a Re-Pull by EmployerTPASOAPTransactionNumber, and a Re-Pull by a Date Range.

<!-- Query element for employer to collect claim responses they are expecting -->

#### REDACTED

For the regular Pull, the caller needs to supply only the Integrated Partner Unique ID. Although there are different ways to verify the calling Integrated Partner besides this element, the Broker uses it as an additional security check. Also, there is a requirement in WSDL 1.1 that a WSDL definition have at least one input attribute.

#### REDACTED

```
<UniqueID>BR000000001</UniqueID>
```

For the Re-Pull by EmployerTPASOAPTransactionNumber, the caller needs to supply the Integrated Partner Unique ID and the EmployerTPASOAPTransactionNumber element out of the **REDACTED**. This will allow the Broker to send the file defined by the EmployerTPASOAPTransactionNumber.

#### REDACTED

```
<UniqueID>BR000000001</UniqueID>
<EmployerTPAAdditionalFactFindingRequestCollectionQueryCriteria>
  <EmployerTPASOAPTransactionNumber>26151</EmployerTPASOAPTransactionNumber>
</EmployerTPAAdditionalFactFindingRequestCollectionQueryCriteria>
```

For the Re-Pull by date range, the caller needs to supply the Integrated Partner Unique ID and the **REDACTED** element out of the **REDACTED**. The Re-Pull by date range will pull all the files that were pulled by the Connector during the date range specified.

The **REDACTED** is a complex query type that is defined as a begin date (BrokerRecordEffectiveDateFrom), an end date (BrokerRecordEffectiveDateTo) and an EmployerTPASOAPTransactionNumber.

The first time this operation is called, the EmployerTPASOAPTransactionNumber must not be included and the date range that the files to be Re-Pulled are included.

When the Broker sends back the first file in this date range, it will include in the SOAP header the next EmployerTPASOAPTransactionNumber that it sent during that date range (in element name NextEmployerTPASOAPTransactionNumber). In the next call to this operation, the caller must include the NextEmployerTPASOAPTransactionNumber as the EmployerTPASOAPTransactionNumber along with the date range. This differentiates to the Broker the next call in the series from a brand new Re-Pull by Date request.

When the Central Broker determines that it has no more files to send back to the Connector in the given date range, the last file sent back to the Connector is indicated by the Central Broker not including the next EmployerTPASOAPTransactionNumber (so there will not be a NextEmployerTPASOAPTransactionNumber included in the http response SOAP header).

**WARNING:** If the end date in the Re-Pull by date range is in the future, this will cause the **Central Broker** to resend all transactions *including* all the resent transactions that the **Central Broker** has been delivering due to this call, thus putting your Connector into an infinite loop until that date is reached. This will tax the Connector and the **Central Broker** needlessly and must be avoided.

First Call:

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>  
<EmployerTPAAdditionalFactFindingRequestCollectionQueryCriteria>  
<BrokerRecordEffectiveDateFrom>2019-01-01T12:00:00.000-  
04:00</BrokerRecordEffectiveDateFrom>  
<BrokerRecordEffectiveDateTo>2019-08-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>  
</EmployerTPAAdditionalFactFindingRequestCollectionQueryCriteria>
```

Subsequent Calls:

**REDACTED**

```
<UniqueID>BR000000001</UniqueID>  
<EmployerTPAAdditionalFactFindingRequestCollectionQueryCriteria>  
<EmployerTPASOAPTransactionNumber>12345678901234567890123456789012</Employer  
TPASOAPTransactionNumber>  
<BrokerRecordEffectiveDateFrom>2019-01-01T12:00:00.000-  
04:00</BrokerRecordEffectiveDateFrom>
```



```
<BrokerRecordEffectiveDateTo>2019-08-31T12:00:00.000-04:00</BrokerRecordEffectiveDateTo>
</EmployerTPAAdditionalFactFindingRequestCollectionQueryCriteria>
```

#### 4.4.6.2.2.2 Central Broker Response to Integrated Partner

When the Broker receives a "Pull" request from an Integrated Partner, it begins assembling all the requests that are intended for that Integrated Partner. It constructs a SOAP message according to the rules for an Additional Fact-Finding SOAP message (less than 8MB, one Integrated Partner per message, etc.). It adds two additional fields to the Additional Fact-Finding Request - the BrokerRecordEffectiveDate and the BrokerRecordTransactionNumber. The BrokerRecordEffectiveDate indicates the date that the request was accepted into the Broker. The BrokerRecordTransactionNumber is a unique record tracking number and must be returned on the response for this record. It then sends the Additional Fact-Finding requests in the HTTP response.

#### REDACTED

```
<EmployerTPAAdditionalFactFindingRequest>
<StateRequestRecordGUID>11000000000000000000000000000001</StateRequestRecordGUID>
  <SSN>000110001</SSN>
  <ClaimEffectiveDate>2019-09-10</ClaimEffectiveDate>
  <ClaimNumber>1231</ClaimNumber>
  <StateEmployerAccountNbr>179424876</StateEmployerAccountNbr>
  <EmployerName>Pennies Inc</EmployerName>
  <FEIN>123456789</FEIN>
  <TypeofClaimCode>1</TypeofClaimCode>
  <BenefitYearBeginDate>2016-09-28</BenefitYearBeginDate>
  <RequestingStateAbbreviation>CA</RequestingStateAbbreviation>
  <UIOfficeName>BU1</UIOfficeName>
  <UIOfficePhone>1234567890</UIOfficePhone>
  <UIOfficeFax>1234567891</UIOfficeFax>
  <AdjudicatorIdentifier>ADJ109</AdjudicatorIdentifier>
  <AdjudicatorName>Linda Hoffman</AdjudicatorName>
  <AdjudicatorPhone>1234567890</AdjudicatorPhone>
  <AdjudicatorEmail>lindh@nomail.com</AdjudicatorEmail>
  <ClaimantLastName>Ron</ClaimantLastName>
  <OtherLastName>OtherLastName1</OtherLastName>
  <ClaimantFirstName>Smith</ClaimantFirstName>
  <ClaimantMiddleInitial>S</ClaimantMiddleInitial>
  <ClaimantSuffix>Jr</ClaimantSuffix>
  <FailureToRespondMessage>FailureToRespondMessage1</FailureToRespondMessage>
  <RequestType>2</RequestType>
  <OtherExchangeBRTN>2592852</OtherExchangeBRTN>
  <NumberOfInquiries>2</NumberOfInquiries>
  <InquiryOccurrence>
    <InquiryNumber>01</InquiryNumber>
    <InquiryQuestion>Why did you not work all available hours?</InquiryQuestion>
```

```

</InquiryOccurrence>
<InquiryOccurrence>
  <InquiryNumber>02</InquiryNumber>
  <InquiryQuestion>Who was your manager?</InquiryQuestion>
</InquiryOccurrence>
<EvidenciaryAttachmentOccurrence>
  <DescriptionofAttachmentCode>3</DescriptionofAttachmentCode>
  <TypeofDocument>TypeofDocument1</TypeofDocument>
  <AttachmentSize>1</AttachmentSize>
  <AttachmentData>MA==</AttachmentData>
</EvidenciaryAttachmentOccurrence>
<RequestDate>2019-09-10</RequestDate>
<ResponseDueDate>2019-09-21</ResponseDueDate>
<FormNumber>006B</FormNumber>
<BrokerRecordTransactionNumber>13155</BrokerRecordTransactionNumber>
<BrokerRecordEffectiveDate>2019-08-28T17:04:08.000-04:00</BrokerRecordEffectiveDate>
</EmployerTPAAdditionalFactFindingRequest>

```

#### 4.4.6.2.2.3 Integrated Partner Acknowledgement to Central Broker

The Integrated Partner **REDACTED** is initiated once the Integrated Partner has received its file from the Broker. The acknowledgement must accompany every Integrated Partner Pull request, as this is the manner in which the Broker knows that the Integrated Partner Pull was successful. This is required even if the Broker has sent back an empty file and a MessageCode of 2. If this is not sent back to the Broker, the next “Pull” call to the Broker will result in the same file being passed back. The Broker will not move on to the next file until it receives a successful acknowledgement. If the Broker receives 3 unanswered Pull requests, it will suspend any processing of Pull requests by the Integrated Partner until the NASWA SIDES team and Integrated Partner Administrator can work out the problem.

The key field in this message is the EmployerTPASOAPTransmissionNumber, which must correspond with the EmployerTPASOAPTransactionNumber sent back in the Broker Response. The remainder of the message is just reporting information; the values are not used for anything at this time. If the Integrated Partner does not collect this information, just return 0 for the number of records and place a valid date in the date fields.

#### **REDACTED**

```

<EmployerTPASOAPTransmissionNumber>3211</EmployerTPASOAPTransmissionNumber>
<NumberOfRequestRecordsReceived>4</NumberOfRequestRecordsReceived>
<NumberOfRequestRecordsInError>0</NumberOfRequestRecordsInError>
<DateStartedReceivingTransmission>2019-08-31T12:00:00.000-
04:00</DateStartedReceivingTransmission>
<DateFinishedReceivingTransmission>2019-08-31T12:00:00.000-
04:00</DateFinishedReceivingTransmission>

```

## 4.5 SOAP Action

From the W3C, the SOAPAction component in SOAP 1.1 is defined as follows:

“The SOAPAction HTTP request header field can be used to indicate the intent of the SOAP HTTP request. The value is a URI identifying the intent. SOAP places no restrictions on the format or specificity of the URI or that it is resolvable. An HTTP client **MUST** use this header field when issuing a SOAP HTTP Request.”

The SOAPAction **MUST** be specified in the SOAP message. The particular SOAP Action required for each message is specified in the WSDL.

There are six SOAP Actions that are defined in SIDES for Separation Information:

1. postStateSeparationRequestCollection
2. pullStateSeparationResponseCollection
3. pullStateSeparationResponseCollectionAcknowledgement
4. postEmployerTPASeparationResponseCollection
5. pullEmployerTPASeparationRequestCollection
6. pullEmployerTPASeparationRequestCollectionAcknowledgement

There are six SOAP Actions that are defined in SIDES for Earnings Verification:

1. postStateEarningsVerificationRequestCollection
2. pullStateEarningsVerificationResponseCollection
3. pullStateEarningsVerificationResponseCollectionAcknowledgement
4. postEmployerTPAEarningsVerificationResponseCollection
5. pullEmployerTPAEarningsVerificationRequestCollection
6. pullEmployerTPAEarningsVerificationRequestCollectionAcknowledgement

There are six SOAP Actions that are defined in SIDES for Monetary & Potential Charges:

1. postStateMonetaryChargeRequestCollection
2. pullStateMonetaryChargeResponseCollection
3. pullStateMonetaryChargeResponseCollectionAcknowledgement
4. postEmployerTPAMonetaryChargeResponseCollection
5. pullEmployerTPAMonetaryChargeRequestCollection
6. pullEmployerTPAMonetaryChargeRequestCollectionAcknowledgement

There are six SOAP Actions that are defined in SIDES for Determinations & Decisions:

1. postStateDeterminationDecisionRequestCollection
2. pullStateDeterminationDecisionResponseCollection
3. pullStateDeterminationDecisionResponseCollectionAcknowledgement
4. postEmployerTPADeterminationDecisionResponseCollection
5. pullEmployerTPADeterminationDecisionRequestCollection
6. pullEmployerTPADeterminationDecisionRequestCollectionAcknowledgement

There are six SOAP Actions that are defined in SIDES for Benefit Charges:

1. postStateBenefitChargesRequestCollection
2. pullStateBenefitChargesResponseCollection
3. pullStateBenefitChargesResponseCollectionAcknowledgement
4. postEmployerTPABenefitChargesResponseCollection
5. pullEmployerTPABenefitChargesRequestCollection
6. pullEmployerTPABenefitChargesRequestCollectionAcknowledgement

There are six SOAP Actions that are defined in SIDES for Additional Fact-Finding:

1. postStateAdditionalFactFindingRequestCollection
2. pullStateAdditionalFactFindingResponseCollection
3. pullStateAdditionalFactFindingResponseCollectionAcknowledgement
4. postEmployerTPAAdditionalFactFindingResponseCollection
5. pullEmployerTPAAdditionalFactFindingRequestCollection
6. pullEmployerTPAAdditionalFactFindingRequestCollectionAcknowledgement

## 4.6 WSDL

The Web Services Description Language (WSDL) is an XML-based language that provides a model for describing Web services. WSDL defines an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the details involved in applications communication.

The WSDL for SIDES is broken into two files per Standard Format. **REDACTED**.wsdl for Separation Information, **REDACTED**.wsdl for Earnings Verification, **REDACTED**.wsdl for Monetary & Potential Charges, **REDACTED**.wsdl for Determinations & Decisions, **REDACTED**.wsdl for Benefit Charges, and **REDACTED**.wsdl for Additional Fact-Finding describe the interfaces exposed to the States from the Broker. **REDACTED**.wsdl for Separation Information, **REDACTED**.wsdl for Earnings Verification, **REDACTED**.wsdl for Monetary & Potential Charges, **REDACTED**.wsdl for Determinations & Decisions for Determinations & Decisions, **REDACTED**.wsdl for Benefit Charges, and **REDACTED**.wsdl for Additional Fact-Finding describe the interfaces exposed to the Integrated Partner from the Broker. See the full WSDL below for a complete description.

- State Separation Information WSDL:  
**REDACTED**
- Integrated Partner Separation Information WSDL:  
**REDACTED**
- State Earnings Verification WSDL:  
**REDACTED**
- Integrated Partner Earnings Verification WSDL:  
**REDACTED**
- State Monetary & Potential Charges WSDL:  
**REDACTED**
- Integrated Partner Monetary & Potential Charges WSDL:  
**REDACTED**
- State Determinations & Decisions WSDL:  
**REDACTED**
- Integrated Partner Determinations & Decisions WSDL:  
**REDACTED**
- State Benefit Charges WSDL:

**REDACTED**

- Integrated Partner Benefit Charges WSDL:

**REDACTED**

- State Additional Fact-Finding WSDL:

**REDACTED**

- Integrated Partner Additional Fact-Finding WSDL:

**REDACTED**

There is no Policy information contained within these WSDLs. This was done for interoperability reasons. If a Connector is using a technology that requires the use of Policy information, a WSDL has been provided within the JAX-WS Model Connector.

#### 4.6.1 WSDL XSD

Along with each WSDL, there are XSD files that define the elements in the WSDL. These schema files are **REDACTED**.xsd and **REDACTED**.xsd for Separation Information, **REDACTED**.xsd and **REDACTED**.xsd for Earnings Verification, **REDACTED**.xsd and **REDACTED**.xsd for Monetary & Potential Charges, **REDACTED**.xsd and **REDACTED**.xsd for Determinations & Decisions, **REDACTED**.xsd and **REDACTED**.xsd for Benefit Charges, and **REDACTED**.xsd and **REDACTED**.xsd for Additional Fact-Finding. There is one support file - **REDACTED**.xsd – used by all exchanges. The XSDs for the current data elements can be found at:

- Separation Information

**REDACTED**

- Earnings Verification

**REDACTED**

- Monetary & Potential Charges

**REDACTED**

- Determinations & Decisions

**REDACTED**

- Benefit Charges

**REDACTED**

- Additional Fact-Finding

**REDACTED**

- Common Elements

**REDACTED**

One XSD file, **REDACTED**.xsd (**REDACTED**.xsd, **REDACTED**.xsd, **REDACTED**.xsd, **REDACTED**.xsd, **REDACTED**.xsd), is used to include other XSD files in the system and it does not contain any additional information. This file is required due to a problem accessing the **REDACTED** namespace in multiple files within the Java libraries used in SIDES. The combined xsd file is used internally by the Central Broker to allow XSD checks to take place on all the SOAP messages and records sent in by the Connectors.

The combined xsd file may be used by connector software, but it is not necessary if the technology and libraries used in the Connectors' implementation do not require it. The combined xsd file can be found at:

**REDACTED**

## **4.6.2 State WSDL**

### **4.6.2.1 State Post WSDL**

#### **4.6.2.1.1 Separation Information State Post WSDL**

The WSDL for the State Post operation in Separation Information is defined as

**REDACTED**

The input (HTTP request) to this operation is a StateSeparationRequestCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateSeparationRequestCollectionAcknowledgement as defined in the **REDACTED**.xsd.

#### **4.6.2.1.2 Earnings Verification State Post WSDL**

The WSDL for the State Post operation in Earnings Verification is defined as

< **REDACTED**

The input (HTTP request) to this operation is a StateEarningsVerificationRequestCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateEarningsVerificationRequestCollectionAcknowledgement as defined in the **REDACTED**.xsd.

#### **4.6.2.1.3 Monetary & Potential Charges State Post WSDL**

The WSDL for the State Post operation in Monetary & Potential Charges is defined as

**REDACTED**

The input (HTTP request) to this operation is a StateMonetaryChargeRequestCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateMonetaryChargeRequestCollectionAcknowledgement as defined in the **REDACTED**.xsd.

#### **4.6.2.1.4 Determinations & Decisions State Post WSDL**

The WSDL for the State Post operation in Determinations & Decisions is defined as

**REDACTED**

The input (HTTP request) to this operation is a StateDeterminationDecisionRequestCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateDeterminationDecisionRequestCollectionAcknowledgement as defined in the **REDACTED**.xsd.

#### **4.6.2.1.5 Benefit ChargesState Post WSDL**

The WSDL for the State Post operation in Benefit Charges is defined as

**REDACTED**

The input (HTTP request) to this operation is a StateBenefitChargesRequestCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateBenefitChargesRequestCollectionAcknowledgement as defined in the **REDACTED**.xsd.

#### **4.6.2.1.6 Additional Fact-Finding State Post WSDL**

The WSDL for the State Post operation in Additional Fact-Finding Information is defined as

**REDACTED**

The input (HTTP request) to this operation is a StateAdditionalFactFindingRequestCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateAdditionalFactFindingRequestCollectionAcknowledgement as defined in the **REDACTED**.xsd.



#### **4.6.2.2 State Pull WSDL**

##### **4.6.2.2.1 Separation Information State Pull WSDL**

The WSDL for the State Pull operation in Separation Information is defined as:

**REDACTED**

The input (HTTP request) to this operation is a StateSeparationResponseCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateSeparationResponseCollection as defined in the **REDACTED**.xsd.

##### **4.6.2.2.2 Earnings Verification State Pull WSDL**

The WSDL for the State Pull operation in Earnings Verification is defined as:

**REDACTED**

The input (HTTP request) to this operation is a StateEarningsVerificationResponseCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateEarningsVerificationResponseCollection as defined in the **REDACTED**.xsd

##### **4.6.2.2.3 Monetary & Potential Charges State Pull WSDL**

The WSDL for the State Pull operation in Monetary & Potential Charges Information is defined as:

**REDACTED**

The input (HTTP request) to this operation is a StateMonetaryChargeResponseCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateMonetaryChargeResponseCollection as defined in the **REDACTED**.xsd.

##### **4.6.2.2.4 Determinations & Decisions State Pull WSDL**

The WSDL for the State Pull operation in Determinations & Decisions Information is defined as:

**REDACTED**

The input (HTTP request) to this operation is a StateDeterminationDecisionResponseCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateDeterminationDecisionResponseCollection as defined in the **REDACTED**.xsd.

#### **4.6.2.2.5 Benefit Charges State Pull WSDL**

The WSDL for the State Pull operation in Benefit Charges Information is defined as:

**REDACTED**

The input (HTTP request) to this operation is a StateBenefitChargesResponseCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is a StateBenefitChargesResponseCollection as defined in the **REDACTED**.xsd.

#### **4.6.2.2.6 Additional Fact-Finding State Pull WSDL**

The WSDL for the State Pull operation in Additional Fact-Finding Information is defined as:

**REDACTED**

The input (HTTP request) to this operation is a StateAdditionalFactFindingResponseCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is a State AdditionalFactFindingResponseCollection as defined in the **REDACTED**.xsd.

### **4.6.3 Integrated Partner WSDL**

#### **4.6.3.1 Integrated Partner Post WSDL**

##### **4.6.3.1.1 Separation Information Integrated Partner Post WSDL**

The WSDL for the Integrated Partner Post operation in Separation Information is defined as:

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPASeparationResponseCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPASeparationResponseCollectionAcknowledgement as defined in the **REDACTED**.xsd.

##### **4.6.3.1.2 Earnings Verification Integrated Partner Post WSDL**

The WSDL for the Integrated Partner Post operation in Earnings Verification is defined as:

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPAEarningsVerificationResponseCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPAEarningsVerificationResponseCollectionAcknowledgement as defined in the **REDACTED**.xsd.

#### **4.6.3.1.3 Monetary & Potential Charges Integrated Partner Post WSDL**

The WSDL for the Integrated Partner Post operation in Monetary & Potential Charges is defined as:

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPAMonetaryChargeResponseCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPAMonetaryChargeResponseCollectionAcknowledgement as defined in the **REDACTED**.xsd.

#### **4.6.3.1.4 Determinations & Decisions Integrated Partner Post WSDL**

The WSDL for the Integrated Partner Post operation in Determinations & Decisions is defined as:

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPADeterminationDecisionResponseCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPADeterminationDecisionResponseCollectionAcknowledgement as defined in the **REDACTED**.xsd

#### **4.6.3.1.5 Benefit Charges Integrated Partner Post WSDL**

The WSDL for the Integrated Partner Post operation in Benefit Charges is defined as:

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPABenefitChargesResponseCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPABenefitChargesResponseCollectionAcknowledgement as defined in the **REDACTED**.xsd.

#### **4.6.3.1.6 Additional Fact-Finding Integrated Partner Post WSDL**

The WSDL for the Integrated Partner Post operation in Additional Fact-Finding is defined as:

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPAAdditionalFactFindingResponseCollection as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPAAdditionalFactFindingResponseCollectionAcknowledgement as defined in the **REDACTED**.xsd.

#### **4.6.3.2 Integrated Partner Pull WSDL**

##### **4.6.3.2.1 Separation Information Integrated Partner Pull WSDL**

The Integrated Partner Pull operation for Separation Information is defined as **REDACTED**

The input (HTTP request) to this operation is an EmployerTPASeparationRequestCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPASeparationRequestCollection as defined in the **REDACTED**.xsd.

##### **4.6.3.2.2 Earnings Verification Integrated Partner Pull WSDL**

The Integrated Partner Pull operation for Earnings Verification is defined as **REDACTED**

The input (HTTP request) to this operation is an EmployerTPAEarningsVerificationRequestCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPAEarningsVerificationRequestCollection as defined in the **REDACTED**.xsd.

##### **4.6.3.2.3 Monetary & Potential Charges Information Integrated Partner Pull WSDL**

The Integrated Partner Pull operation for Monetary & Potential Charges Information is defined as

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPAMonetaryChargeRequestCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPAMonetaryChargeRequestCollection as defined in the **REDACTED**.xsd.

#### **4.6.3.2.4 Determinations & Decisions Integrated Partner Pull WSDL**

The Integrated Partner Pull operation for Determinations & Decisions is defined as

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPADeterminationDecisionRequestCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPADeterminationDecisionRequestCollection as defined in the **REDACTED**.xsd.

#### **4.6.3.2.5 Benefit Charges Integrated Partner Pull WSDL**

The Integrated Partner Pull operation for Benefit Charges is defined as

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPABenefitChargesRequestCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPABenefitChargesRequestCollection as defined in the **REDACTED**.xsd.

#### **4.6.3.2.6 Additional Fact-Finding Integrated Partner Pull WSDL**

The Integrated Partner Pull operation for Additional Fact-Finding is defined as

**REDACTED**

The input (HTTP request) to this operation is an EmployerTPAAdditionalFactFindingRequestCollectionQuery as defined in **REDACTED**.xsd.

The output (HTTP response) is an EmployerTPAAdditionalFactFindingRequestCollection as defined in the **REDACTED**.xsd.

**5. BUILD THE CONNECTOR: SECURING THE MESSAGE**  
**REDACTED**

## 6. CONNECT WITH THE CENTRAL BROKER: SENDING THE MESSAGE

### 6.1 Sending a message

The destination address for all messaging with the Central Broker for the Separation Information exchange is:

**REDACTED**

The destination address for all messaging with the Central Broker for the Earnings Verification exchange is:

**REDACTED**

The destination address for all messaging with the Central Broker for the Monetary & Potential Charges exchange is:

**REDACTED**

The destination address for all messaging with the Central Broker for the Determinations & Decisions exchange is:

**REDACTED**

The destination address for all messaging with the Central Broker for the Benefit Charges exchange is:

**REDACTED**

The destination address for all messaging with the Central Broker for the Additional Fact-Finding exchange is:

**REDACTED**

### 6.2 Sample SOAP message sent

Below is a sample of a State SOAP message that a Connector has sent to the Broker for Separation Information:

Sample State SOAP Message

**REDACTED**

### 6.3 Acknowledgements

The acknowledgement in any of the message types is an important part of the transaction. They let the receiver know that the file has successfully made it to its destination, regardless of the individual records success or failure. If an acknowledgement is not received within 15 minutes of sending a message, failure must be assumed and the message should be sent again.

A message failure and re-send could be an infinite cycle if one of the Connectors is having difficulty. Therefore, Connectors must be implemented to limit this process to occur only three times (an initial time plus two retries) in an

automated fashion before contacting the Connector's SIDES administrator. The State or Integrated Partner administrator can then begin debugging the problem. If the problem was noted by the Central Broker, the NASWA SIDES team will be notified and error resolution will be started on the Central Broker side also.

#### **6.4 Non-Broker Returns**

There are some messages that a State, or Integrated Partner Connector may receive when attempting to communicate with the Central Broker that are non-standard Central Broker return messages. These messages were not discussed earlier in this document as part of the message exchange with the Central Broker as they are not message related but, rather, are overall system-related return messages. These must be handled by all Central Broker client connectors.

1. "UI SIDES Server Error has occurred. Please try again in a few minutes or contact your UI SIDES Administrator for assistance."

**REDACTED**

2. A no response. A [404] Http Error.

**REDACTED**



## 7. CONNECT WITH THE CENTRAL BROKER: TESTING CONNECTOR SOFTWARE

The connector software must be designed, coded and thoroughly tested to ensure correct functionality when interfacing with the Central Broker. Connector testing responsibilities are highlighted as well as key points for consideration when testing your connector software. To support connector testing, a set of tools are available to aid in the development and testing process.

### 7.1 Connector Responsibility

It is the State or Integrated Partner's responsibility to fully test their SIDES implementation. Connector testing includes the backend system components, system interfaces, and the connector software that interacts with the Central Broker. The Connector's process for testing is not prescribed as different organizations must follow their own testing procedures and standards. Prior to production operations, the Connector must perform the Self-Certification Readiness Review (see Section 8).

This Developer Guide not only provides developers with a roadmap for constructing the connector, but also serves as a tool the test team may use to support comprehensive testing of the connector software. The following list highlights key points to be considered when testing your connector software:

- Creation of the Request or Response XML
  - All the data required in the Standard Format is being accessed correctly
- The XML data format (Standard Format)
  - XSD violations of the Standard Format must be trapped and handled on the Connector side.
  - For a State using the Separation Information Standard Format, all the ClaimantSepReasonCodes have been tested along with any business rule dependencies on the individual ClaimantSepReasonCode.
  - For the Integrated Partners using the Separation Information Standard Format, all the EmployerSepReasonCodes have been tested along with any business rule dependencies on the individual EmployerSepReasonCode.
  - Other values that are called out in the Standard Format. A test case should not only be made for the negative (error) conditions, but also the positive (non error) conditions – especially around the boundaries.
  - The entire set of error codes, which result from business rule violations, must be handled. This must be performed to cover the case where an error code is received internally before the message is generated and

after the message is sent and the error code comes back in the acknowledgment.

- SOAP concepts
  - All of the required custom headers are included with the message.
  - The SOAP action is part of the https message.
  - Generation of the digital signature and encryption occurs correctly.
  - The message timestamp has a **REDACTED** time to live.
- Message Codes
  - Central Broker generated message codes on the acknowledgment of a Post and an http response on the Pull.
  - Connector generated message codes on the acknowledgement of a Pull.
- Duplicate records
  - Duplicate records Pulled in the same file and in a different file.

## **8. CONNECT WITH THE CENTRAL BROKER: SELF-CERTIFICATION READINESS REVIEW**

This section provides details and discussion on the final testing process where the Connector must demonstrate it is ready to interface with the production Central Broker. This is a critical section as the Connector must perform the Self-Certification Readiness Review before being allowed to enter operations with SIDES.

### **8.1 Production Readiness**

The Self-Certification Readiness Review is initiated after the Connector fully completes software development, internal testing, user acceptance testing (UAT), and round-trip testing with a State or Integrated Partner.

Prior to production operations, the State or Integrated Partner must self-certify that repulls by date and by SOAPTransactionNumber can be performed, ensure the network connection remains active for up to 15 minutes, and the Connector's system handles the defined set of business rule error codes specified in Part B of the Developer Guide. Business rule error codes correspond to the business rules defined in the data exchange standard format.

A Connector may use different technologies and programming languages to create their client program, therefore the Self-Certification Readiness Review does not inspect the client source code or design, but it relies on the Connector to fully test and validate its software.

Connectors must create their own dataset to test each business rule and they should work with their network engineers to obtain documentation that confirms the network timeouts are configured per SIDES requirements.

The Self-Certification document for States and Integrated Partners can be found on the SIDES Website at <https://info.usides.org>. Below is a snippet of the document that must be fully filled out and signed by your agency's SIDES representative providing attestation that all SIDES requirements have been implemented and tested.

Email the completed document to the NASWA SIDES team at [sides-support@naswa.org](mailto:sides-support@naswa.org). The NASWA SIDES team will review the documentation, seek clarification if necessary, and certify the Connector for production operations.

**STATE AGENCY NAME:**

If applicable, name of vendor working on SIDES projects: \_\_\_\_\_

**REPULLS:**

Test Case Description	Executed by:	Pass/Fail	Comments
Re-pull by date		<input type="checkbox"/> Yes <input type="checkbox"/> No	
Re-pull by <u>StateSOAPTransactionNumber</u>		<input type="checkbox"/> Yes <input type="checkbox"/> No	

**NETWORK TIMEOUT**

Test Case Description	Confirmation Provided by:	Confirmation	Comments
Network hardware timeouts will be set to ensure SIDES sessions (active or idle) will remain open for up to 15 minutes to permit SIDES transactions ample time to complete.		<input type="checkbox"/> Yes <input type="checkbox"/> No	

**STANDARD FORMAT BUSINESS RULE ERROR CODE HANDLING**

Indicate who executed the test case, whether the error code was successfully detected and trapped, and supply any comments/explanation for error codes that are not handled by the Agency's connector.

Error Code	Test Case Description	Executed by:	Pass/Fail	Error Code Validation Source	Comments
2002	Business Rule violation - Two or more <u>UniqueAttachmentIDs</u> assigned to a specific Separation Information Request are the same, they must be unique.		<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> User interface <input type="checkbox"/> Database logs <input type="checkbox"/> Model Connector logs <input type="checkbox"/> Non-Model Connector logs	
2003	Business Rule violation - Null data or whitespace in field.		<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> User interface <input type="checkbox"/> Database logs <input type="checkbox"/> Model Connector logs <input type="checkbox"/> Non-Model Connector logs	
2004	Business Rule violation - Date with greater than 4 digit year.		<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> User interface <input type="checkbox"/> Database logs <input type="checkbox"/> Model Connector logs <input type="checkbox"/> Non-Model Connector logs	
2010	Business Rule violation - PIN (A-34) can only be used if the file the record is in is		<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> User interface <input type="checkbox"/> Database logs <input type="checkbox"/> Model Connector logs	

**CERTIFICATION**

\_\_\_\_\_ certifies, the SIDES functionality and test cases have been fully implemented and successfully executed

For Broker Endpoint Name \_\_\_\_\_ Broker Number \_\_\_\_\_

Signed By: \_\_\_\_\_

Date: \_\_\_\_\_

**8.1.1 Production Readiness Information**

The Self-Certificatin Readiness Review is a required step that must be performed prior to production operations or after any major change to the Connector's system. It is important that the connector software be properly vetted so that valuable production time is not spent on items that could have been prevented during testing.



Production readiness approval of the Connector is achieved when:

- Re-pulls by date range and SOAPTransactionNumber work
- Network connections are set to remain open even if they are idle for **REDACTED**
- All business rule error codes are trapped by the Connector's software

The following process needs to be followed before a Connector will be allowed to join SIDES in production.

- Step 1 - Download Self-Certification document from the SIDES Website at <https://info.usides.org>.
- Step 2 – Conduct testing
  - Step 2.1 – Using the Self-Certification document, develop test data for each business rule error code that must be handled, and create test scenarios for re-pull testing.
  - Step 2.2 – Run each test case and compare the results obtained from your test system to ensure error codes are trapped on the Connector system and re-pulls provide the expected request/response files.
  - Step 2.3 - Fill in the Self-Certification document with the results of the test, and sign it attesting that all SIDES requirements have been implemented and tested.
- Step 3 – Email the completed document to the NASWA SIDES team at [sides-support@naswa.org](mailto:sides-support@naswa.org).

#### **8.1.1.1 Step 1 - Download Self-Certification Documentation**

The Self-Certification documents can be found on the SIDES Website at <https://info.usides.org>. Navigate to the Production Self-Certification Review section and download the document corresponding to the SIDES data exchange being certified. For example, files for the separation information exchange are:

- **REDACTED**

#### **8.1.1.2 Step 2 – Conduct Testing**

##### **8.1.1.2.1 Step 2.1 – Develop Test Data and Test Scenario**

The Self-Certification Readiness Review document is your Connector's test plan from which test data must be developed for each error code that can be

returned by the Central Broker. Make sure to retain test data files to repeat tests until all tests are passed.

For re-pulls, States must develop their own test scenarios that identify the specific response records to be returned by a re-pull by date range. The expected response record to be returned for a re-pull by SOAPTransactionNumber must be identified prior to running the test case.

Likewise, Integrated Partners must define test scenarios where the expected requests are defined for the re-pull by date range. For re-pull by SOAPTransactionNumber, the specific request to be returned by the re-pull must be identified prior to the test.

#### **8.1.1.2.2 Step 2.2 – Run Test**

The bulk of the Self-Certification Readiness Review consists of testing business rule error codes. Business rule error code testing ensures that bad data is filtered out by the Connector prior the transmission to the Central Broker, thus ensuring the integrity of the SIDES production system and optimizing Central Broker performance.

All testing by the State and Integrated Partner must be directed to the SIDES test system Central Broker. No production data may be sent to the test database.

For standard format business rule error code testing, each test case (request file for States and response file for Integrated Partners), is submitted to the Central Broker. This process is completed for all test cases and continues until the tester has successful tests for each error code specified in the Self-Certification document.

Re-pull tests must be performed by the State or Integrated Partner. The tester compares the test results with the expected results defined in the test scenario. Testing of the re-pull functionality continues until the data returned from the Central Broker matches the expected results defined in the test scenario.

#### **8.1.1.2.3 Fill in the Self-Certification document with test results**

After testing is complete, the Self-Certification document must be completed.

Fill out the State Agency Name and if applicable, specify the name of the vendor working on the SIDES project.

Next, fill out the Re-pull section. Indicate who executed the test, if the test passed or failed, and provide details of the test scenario and expected data to be returned by the re-pull. For example, the comments may indicate the re-pull by date range was performed for January 5, 2022 starting at 8AM ET through January 6, 2022 11:59:59ET. The expected results were 5 records with the following record GUIDs. Test results returned 5 records each matching the expected record GUIDs.

Fill out the Network Timeout section of the document. Provide the name of the individual who confirmed the network connection will remain active for up to 15 minutes even if the session is idle. Check the confirmation Yes/No box, and provide the documentation / evidence in the comments area.

Fill out the Standard Format Business Rule Error Code Handling section. Indicate who executed the test, whether the test passed or failed, specify the source where the error code was validated, and fill out the comments field to provide explanation of how the system handles the test case.

Example: The system is coded to populate the request date by using the current date (if request is sent before 5PM) or the next business day (if request is after 5PM). As a result, the test case cannot be confirmed through log files or the user interface, but is confirmed based on inspection of the database and software code.

Fill out the Certification section of the document. Enter the name of the State or Integrated Partner who will provide the self-certification, provide the Endpoint name and Broker Number, sign and date the document.

### **8.1.1.3 Step 3 – Email the Self-Certification Document to the NASWA SIDES Team**

Completion of the Self-Certification Readiness Review requires a signature by your agency's SIDES representative providing attestation that all SIDES requirements have been implemented and tested.

Email the completed Production Readiness Self-Certification document to [sides-support@naswa.org](mailto:sides-support@naswa.org). The NASWA SIDES team will review the document, follow-up with any necessary clarifications, if needed, and certify the State of Integrated Partner for production operations.

## **9. LESSONS LEARNED, COMMON MISTAKES, THINGS TO REMEMBER, KEY DEVELOPMENT PITFALLS**

### **9.1 Lessons Learned**

#### **9.1.1 Using the Black Box jar file in a larger program**

The black box was developed to help those States and Integrated Partners that wanted a quick and easy way to get up on SIDES. The jar file included with it was meant to be called from within its own java container. We learned that some Connectors had actually taken the black box jar file and incorporated the jar into a larger programs lib directory. This had some unintended consequences to do with logging. The Model Connector grabs the root logger and works off of it. But, if this is in a larger program with log4j as its parent logger, it had the unintended consequence of writing out large volumes of logging output as the Model Connector log files get intertwined with the parent systems log file(s). When running requests for E-Response, which can have hundreds or even thousands of Posts, this led to hundreds of GBs worth of log files. The Model Connectors have been modified to work inside a larger java container now independently of another logger.

Although the Model Connector logs have been implemented to not write into other logs, the inverse may also hold true and the other logs may be written into the Model Connector logs. There is nothing the Model Connectors can do to stop this event, therefore the endpoint may need to add an additional parameter to their log configuration. That parameter is called Additivity, and in log4j the call is must be set to 'false'. By default, this value is set to true. Please see the log4j documentation for further information on this parameter.

#### **9.1.2 Using the Black Box jar to call directly to the State/Integrated Partner Post/Pull methods**

When the Model Connectors are released, there is a source code directory and a black box directly. The source code directory was put in so that a State/Integrated Partner could modify the code to call the Post and Pull routines directly using the objects instead of the flat file structure. When doing this, it is assumed that all the setup of loggers and such would be incorporated into that larger program. A problem occurs when Connectors use the black box they don't necessarily see the setup of the loggers and such. If a Connector is going to use the black box jar file yet are calling directly into the Post/Pull routines, they must set up logging before the call and call the destroy loggers after the call. Please see the source code for examples of how to set up the loggers.



## 9.2 Common Mistakes

### 9.2.1 Invalid To: and/or From:

One common mistake is to provide the 'TO:' and 'FROM:' SOAP headers in an incorrect manner. The 'TO:' and 'FROM:' must be in the form described in Section 4.3 SOAP Custom Headers.

Most of the instances of this type seen by the Central Broker are when the two fields have been reversed.

Also, the 'TO:' on Pulls must be to the Broker and not an individual Connector.

### 9.2.2 Connector Not a Participant

The Unique IDs presented in Section 4.2.1 - Unique ID are the only participants SIDES recognizes.

The most common error in this category is using older Unique IDs. (There was a change in the Integrated Partner Unique IDs to accommodate SIDES E-Response.)

Another common error is to omit one or more digits on the Integrated Partner's Unique ID. It will always be a 'BR' followed by nine digits for a SIDES Web services participant (as opposed to an E-Response website participant – which is the nine-digit Federal Employer Identification Number).

### 9.2.3 Invalid SOAP Action

Without a properly defined SOAP Action, the Central Broker does not know how to process the incoming message. **REDACTED**. The State, or Integrated Partner must handle this situation by its HTTP response timeout-handling routine.

### 9.2.4 Incorrect/Missing Security

As discussed in Section 5 - BUILD THE CONNECTOR: securing the message, security is of paramount concern to the Broker. When first beginning the process to connect with the Broker, the hardest problems encountered involve **REDACTED**. The Connector should make sure that it is following all the **REDACTED** preparations for the message. If the Connector believes it is doing everything correctly, then it should get in touch with the NASWA SIDES team to help debug the issue.

### 9.2.5 Central Broker Not Having Up-to-Date **REDACTED** Information

**REDACTED**

### 9.2.6 **REDACTED** on Server Not Accurate

**REDACTED**

### 9.2.7 Interpretation of Standard Format for Money Fields

Numeric fields defined in a standard format use a fixed-point representation of a number whose total length is represented by the digits to the left of the decimal

point and the fractional amount is represented by the digits to the right of the decimal point. For example, the TotalEarnedWages field in Separation Information is defined as numeric 15.2. This represents a number (dollar amount) that can have 13 digits to the left of the decimal point and two digits to the right of the decimal point. The largest number this field can contain is 9,999,999,999,999.99, while the smallest number is 0.00.

### **9.2.8 State Employer Account Number**

As per the SIDES standard formats, UI agencies must pass **REDACTED** to Integrated Partners. The **REDACTED** is a character field with a maximum length of **REDACTED** bytes that the Integrated Partner to locate the separating employer in their automated systems. When implementing SIDES, it is essential that the UI agency does not change the **REDACTED** that the Integrated Partners presently receives. For example, if an Integrated Partner presently receives a **REDACTED** without a location code, please ensure that location code is not appended to the **REDACTED**. Otherwise, the Integrated Partner will not be able to respond to the request for separation information as the **REDACTED** cannot be looked up on their system.

### **9.3 Things to Remember**

This section discusses some of the issues to be considered by a developer beyond the scope of the connection to the Central Broker.

#### **9.3.1 REDACTED**

#### **9.3.2 Existing Business System Modifications**

Key aspects of a State or Integrated Partner connector solution are having a data store that conforms to the standard separation request and response formats, being able to collect the requisite request data from the claimants to populate the request part of the data store, deliver the standard response data collected from an Integrated Partner via the Broker, and integrate imaging/document management technology into the solution.

So, in addition to the communications portion of the connector system, a State Connector must be able to consume and generate XML-based data and interact with the State's end users (adjudicators) in terms of delivery of the data to them for processing. For Separation Information., request data may be collected from claimants as part of an Internet Claim Filing application that includes "intelligent" fact finding such that the appropriate question/data flows are presented based on the separation reason chosen by the claimant, or, alternatively, this separation data may be collected during the fact finding by a Call Center agent using screens or Web pages consistent with the separations request format.

It is very likely that Integrated Partners will provide attachments as part of their responses (for exchanges that include attachments). Therefore States will need to be able to consume these attachments, and organize and route them in accordance with their non-monetary adjudication processes.

Finally, as a template for the user interface for delivery of the separation data to State non-monetary adjudication staff for processing, the SIDES E-Response website can be an aid, as all the applicable question for each response data element has been defined as well as the page flows for each of the possible 22 separation reasons.

If an agency currently has a paper system you will need to consider what to do with the XML data when it arrives with you. In the worst case, it can be printed out and used in the current manner as in your existing process, but with the benefits of better data quality, no postal delays, and no postage costs.

If an agency has an existing electronic system, a State and Integrated Partner will need to make sure that the SIDES XML data is interfaced into the target system successfully.

### **9.3.3 Error Handling**

Discovering errors in the data sent to it is one of the important functions performed by the Central Broker. But having Connectors send records with errors wastes throughput and ties up valuable Central Broker resources. This is one of the reasons why it is important that each Connector implement the business rules/validation on their systems also.

If, by chance, an error is received from the Central Broker, then more than just a correction should be made to that record. A review needs to take place on why that data was allowed to be sent to the Central Broker in the first place, and any corrections made to the Connector's system must be made (or a global change needs to take place with all Connector systems and the Central Broker if an inherent flaw is discovered).

### **9.3.4 Disaster Recovery**

**REDACTED**

## **9.4 Key Development Pitfalls**

**9.4.1 REDACTED**

**9.4.2 REDACTED**

**9.4.3 REDACTED**

**10.LIST OF TABLES**

Table 1 – Exchange Format ..... 14

Table 2 – Separation Information Backfilled Data ..... 40

Table 3 – Earnings Verification Backfilled Data ..... 41

Table 4 - Monetary & Potential Charges Backfilled Data..... 42

Table 5 - Determinations & Decisions Backfilled Data ..... 43

Table 6 – Benefit Charges Backfilled Data ..... 46

Table 7 – Additional Fact-Finding Backfilled Data..... 47

Table 8 - Unique IDs of Current Participating States ..... 52

Table 9 - Unique IDs of Current Participating Integrated Partners..... 53

Table 10 – Post-Acknowledgement Message Codes ..... 54

Table 11 – Pull-Response Message Codes ..... 55

Table 12 - Pull Acknowledgement Codes ..... 55

Table 13 - State Post to Broker ..... 57

Table 14 - State Post to Broker - SIDES Employer Website (E-Response)..... 58

Table 15 – State Post to Broker – SIDES Employer Website (E-Response) ..... 60

Table 16 – Broker Acknowledgement to State Post ..... 60

Table 17 - State Request to Broker (Regular Pull) ..... 61

Table 18 - Broker Response to Request (Regular Pull) ..... 62

Table 19 - State Acknowledgement to Broker (Regular Pull) ..... 62

Table 20 - State Request to Broker (Re-Pull by StateSOAPTransactionNumber) .. 63

Table 21 - Broker Response to Request (Re-Pull by StateSOAPTransactionNumber)  
..... 64

Table 22 -State Acknowledgment to Broker (Re-Pull by  
StateSOAPTransactionNumber) ..... 65

Table 23 - State Request to Broker (Re-Pull by Date) ..... 66

Table 24 - Broker Response to Request (Re-Pull by Date) ..... 67

Table 25 - State Acknowledgement to Broker (Re-Pull by Date)..... 68

Table 26 – Integrated Partner Post to Broker ..... 68

Table 27 - Broker Acknowledgement to Integrated Partner Post..... 69

Table 28 – Integrated Partner Request to Broker (Regular Pull) ..... 70

Table 29 - Broker Response to Request (Regular Pull) ..... 71



Table 30 – Integrated Partner Acknowledgment to Broker (Regular Pull).....71

Table 31 – Integrated Partner Request to Broker (Re-Pull by  
EmployerTPASOAPTransactionNumber) .....72

Table 32 - Broker Response to Request (Re-Pull by  
EmployerTPASOAPTransactionNumber) .....73

Table 33 – Integrated Partner Acknowledgment to Broker (Re-Pull by  
EmployerTPASOAPTransactionNumber) .....74

Table 34 – Integrated Partner Request to Broker (Re-Pull by Date).....76

Table 35 - Broker Response to Request (Re-Pull by Date) .....76

Table 36 – Integrated Partner Acknowledgment to Broker (Re-Pull by Date) .....77