



THE SURPLUS LINE ASSOCIATION OF CALIFORNIA

---

# Automated Batch Submission

---

APRIL 30, 2015



## TABLE OF CONTENTS

1. Introduction	4
2. XML Batch Creation Guidelines	5
2.1. Batch Definition	5
2.2. SLA California Batch Filing Guidelines	5
2.2.1. XML Batch File Size and Structure	7
2.2.2. XML Batch File Name	8
2.2.3. HTML (XML) Encoding	9
2.2.4. Create an XML Batch File	10
2.2.5. Initial XML Batch File Validation	10
2.2.6. Secondary XML Batch File Validation	11
2.2.7. Table of XML Fields	12
2.3. Additional XML Information	30
3. Bulk Upload (XML)	31
3.1. Description	31
3.2. Prerequisites	31
3.3. Process	32
3.3.1. Create Batch File	34
3.3.2. Log in to SLIP	34
3.3.3. Upload and Submit the File	34
3.3.4. SLIP Validates the File	36
3.3.5. Monitor the Batch Submission Status	37
3.3.6. Batch File is Accepted or Rejected	37
4. API Batch Submission (XML)	38
4.1. Description	38
4.2. Pre-requisites	38
4.3. Process	39
4.3.1. Create Batch File	41
4.3.2. Submit Batch File	41

4.3.3. SLIP Validates the File	41
4.3.4. Monitor the Batch Submission Status	41
4.3.5. Review Notifications	42
4.4. Methods	42
4.4.1. Credential Verification Endpoint	43
4.4.2. Upload Batch Filing Endpoint	45
4.4.3. Check Status Endpoint	47
4.4.4. Get Notifications Endpoint	50
5. Batch Image File (BIF) Upload (Non-XML)	53
5.1. Description	53
5.2. Pre-requisites	53
5.3. Process	53
5.3.1. Create New Batch	54
5.3.2. Log in to SLIP	54
5.3.3. Upload and Submit BIF	54
5.3.4. SLA Validates the File	56
6. Frequently Asked Questions	57

## 1. Introduction

SLIP (Surplus Lines Information Portal) is a portal that allows brokerages to submit policy data to SLA California electronically. This document provides instructions for the automated submission options available, through SLIP, to users of the SLA California system.

While brokerages will always be able to submit data manually into SLIP, bulk uploads of batches streamline processes and save time. Brokerages using products that don't integrate with SLIP will have to enter data twice and track submissions separately.

There are three methods for automated bulk uploads via SLIP: Bulk upload (XML), Automated Process Interface (API) submission, and Batch Image File (BIF) upload. The standard way to automate batch submission is to export the policy data as XML so it can be uploaded manually as a single ZIP file (Bulk Upload). You can take automation further by using API submission for complete integration. Finally, you can upload a group of document images as a batch (BIF).

The remainder of this document summarizes these automated batch submission methods and provides the technical details necessary to implement each method.

## 2. XML Batch Creation Guidelines

If you decide to implement either the Bulk Upload or the API batch submission method there are some common guidelines for automated batch submission. This section provides the common guidelines and requirements for these automated batch submission methods. For details on the Batch Image File upload process, please see section 5. below.

### 2.1. Batch Definition

In SLIP, a Batch refers to each group of policies submitted to the SLA for review. Any XML file containing data for one or more policies is considered an XML batch file. The XML batch file is combined with any additional policy documentation (electronic document images of Declaration Pages, Syndicate List, SL Forms, etc.) into a single ZIP file for use in the Bulk Upload method.

### 2.2. SLA California Batch Filing Guidelines

Brokerages should submit filings in batches of no more than 75 documents with a cover sheet. If a brokerage's filings exceed 100 items in a month, the brokerage should file batches weekly or at least twice per month.

The batch cover sheet should include the following information:

- Assigned SLA broker number
- Name of the insured
- Policy number
- Premium amount
- Stamping fee (for each item)
- Total number of items being filed
- The type of document being filed(new, renewal, endorsement, cancellation, extension endorsement, or offset)

Items in the batch cover sheet should be arranged as follows:

- Items should be listed in the same order as the documents in the batch
- The state surplus line tax and stamping fee must be shown separately on the filing document.
- The invoice date must be included for each policy and endorsement filed.
- The date invoiced can be shown on your cover sheet or declarations page.
- A copy of the invoice may be attached to the policy or endorsement.

The documents in a batch should be submitted in the following order:

- New and renewal policies
- Endorsements

- Cancellation endorsements
- Non-money endorsements
- Documents for each new or renewal policy should be submitted in the following order:
  - Declaration page
  - Cover note or binder
  - Forms or attachments to the policy (if applicable)
  - SL-1 form
  - SL-2 form (when applicable)

More information and details about specific types of filings can be found on the SLA California website's Filing Procedures section located online at <http://www.slacal.org/brokers/filing-procedures>.

### 2.2.1. XML Batch File Size and Structure

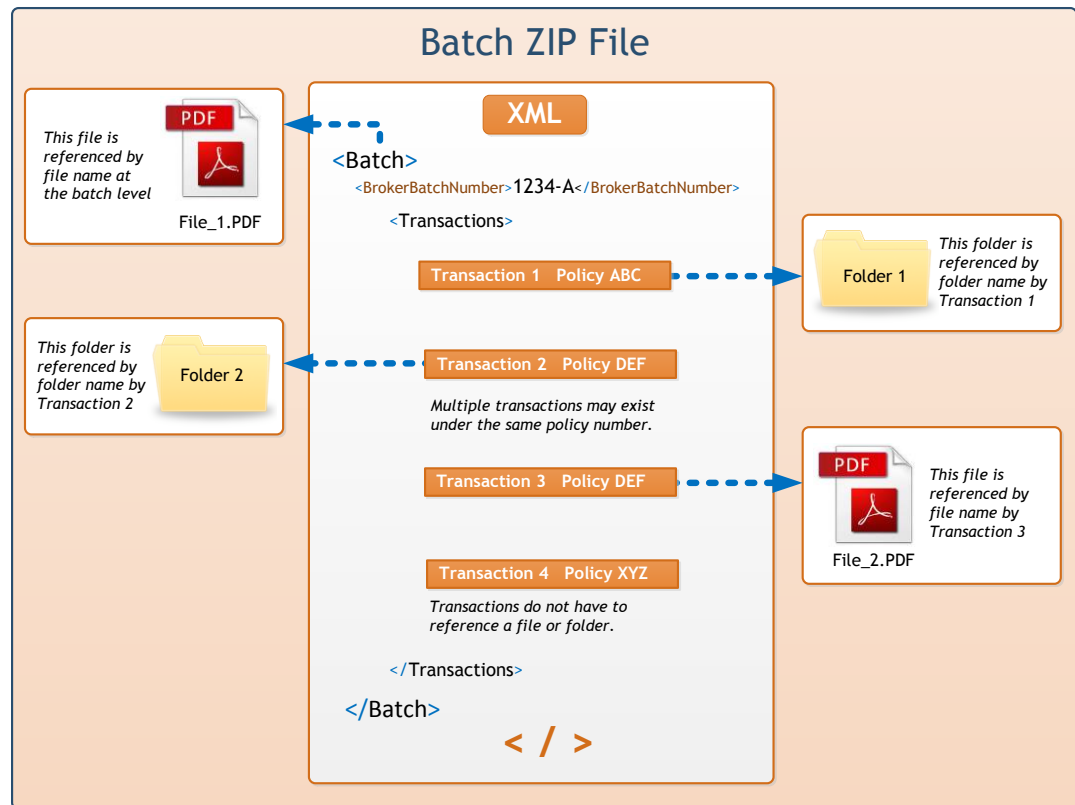
The following guidelines apply to the XML Bulk Upload and the API Batch Submission:

- The batch ZIP file size cannot exceed 1 GB.
- The root of the ZIP file should contain one XML file and the policy documents and/or folders containing policy documents associated with the filings in the XML file. Policy documents and supporting documentation can be associated at the batch level or at the transaction (policy) level within the XML.
- The XML file can contain an element that references individual file names or a folder that can contain one or more files. (These elements in the XML file allow SLIP to associate a file or group of files with either the batch as a whole, or with a specific transaction.) Files that are saved directly in the batch ZIP file (a.k.a. the “root”) can be referenced directly in the XML file by using the file name.
- Files that are saved to a folder within the batch ZIP file must be referenced by the name of the folder in which they are located.

Brokerages can name files or folders associated with the batch according to their own business rules. The XML file can be created to include elements referencing a specific file or folder name for a batch or for a transaction based on these rules at the time the XML file is generated. For example, a brokerage may choose to have a folder created for each transaction. The business rules may state that the folder will be named based on the policy number and transaction identifier (or other identifier relevant to the brokerage management system that allows a user to identify the transaction). In this instance, the XML file would create an element for each transaction where the referenced folder is named with a combination of policy number and transaction ID (e.g., Folder Name = “PolicyNumber\_TransactionID”).

Each brokerage can determine its own process for naming files or folders and modify the system to generate the XML file according to those rules. Selecting a naming convention associated with the policy or transaction identifier may make it easier for the user to identify a document's folder without having to review the XML file to find the element. Each folder should be given a unique name.

The graphic below represents the concepts related to files and folders that are referenced in the XML file:



- The batch ZIP file contains the XML file, and the referenced files or folders.
- There is an element at the batch level that references supporting documentation. This element may reference individual files or a folder.
- There is an element for each transaction within the batch that references supporting documentation. This element may reference individual files or a folder.
- If supporting documentation is referenced by file name, it should be included at the root level (directly inside the batch ZIP file). If files are saved to a folder within the batch, the folder name should be used in the reference.
- Each folder name in the batch ZIP file should be unique (i.e., no two folders can share the same name).

### 2.2.2. XML Batch File Name

The file name is limited to 200 characters. While there is no required naming convention, we recommended that filenames should make maintaining and tracking your submissions easy. We suggest that you include the submission date and time in the file name. For example, 20100501\_0930\_Batch.zip (date\_time\_Batch.zip or CCYYMMDD\_HHMM\_Batch.zip) would indicate the batch was created on 05/01/2010 at 9:30 AM.



### 2.2.3. HTML (XML) Encoding

Several special characters are reserved and cannot be used directly in XML element or attribute data. Replace them with XML Entity references or XML Encoded text. These special characters act as flags to the parser; they delimit the document's actual content and tell the parser to take specific actions. These special characters, therefore, must be represented in their encoded format:

Character Name	Reserved Character	Entity Reference
Ampersand	&	&amp;
Apostrophe	'	&apos;
Quote	"	&quot;
Less Than	<	&lt;
Greater Than	>	&gt;

#### **2.2.4. Create an XML Batch File**

The creation of the batch file will require the involvement of a technical resource that is familiar with XML and the data management system in use by your brokerage. Several different data management systems are used by brokerages throughout the country; therefore, this document cannot provide step-by-step instructions on how to extract policy data from your specific data management system. Rather, this document identifies the structure and formatting requirements of the batch submission in its final form.

The first step in the creation of the batch file is to identify the criteria by which policy data should be extracted from the brokerage's data management system. Typically, brokerages extract data based on a specified date range or some other criteria indicating a submission to SLA California is required.

Once the criteria to extract policy data is identified for your data management system, a technical resource must create the XML file that contains the policy data. For details on the required format and structure of the XML file, please refer to Section 2 – Batch Creation Guidelines. An XML schema will be provided upon request. The XML schema identifies technical constraints on the content and structure of the XML file and can be used to validate the XML file prior to submission.

Once the XML file is created and the policy documents have been identified and/or extracted from the data management system, all of the files should be included in one ZIP file for submission to SLA California.

Note: The system will not accept Microsoft Excel files saved as XML Data or XML Spreadsheet file types. Please follow the XML format described in this document and identified within the XML Schema to create the XML file.

#### **2.2.5. Initial XML Batch File Validation**

This section describes the first set of validations performed during the initial batch submission. If the document fails ANY of the validations identified below, the ENTIRE batch file will NOT be accepted and will not move on to step 2 of the initial validation.

In the initial file validation process, SLIP completes the following steps:

- Parse the document and check that the document is well-formed.
- Check the XML file document against the XSD (XML Schema Definition) file.
- Check the length of all data elements to ensure they do not exceed maximum lengths.
- Check that values of the specified elements comply with the detailed XML document requirements and the XML schema.
- Check that the document file names (policy documentation) contained within the ZIP file match those listed in the XML file.
- Check that all associated policy document types are PDF, PDF/A, PNG, TIFF, or JPG/JPEG. Any other format will not be accepted.

- Check that each associated policy document (contained in the ZIP file) is less than or equal to 150 MB.

If any of the batch data is invalid, the system will reject the entire file. An email will be sent to the SLIP user explaining the rejection and requesting resubmission. The user must correct the batch and resubmit it through SLIP. If the data file is valid, the file will be imported into a temporary queue for policy-level data validation.

#### **2.2.6. Secondary XML Batch File Validation**

This section describes the secondary validation performed by SLIP after the initial validation is passed. If the policy data in the XML file fails ANY of the following validations, the ENTIRE batch file will NOT be accepted.

In the secondary file validation process, SLIP completes the following steps:

1. Check for a valid broker or brokerage identification number. The brokerage identification number is the SLA Broker Number for which the batch is being submitted. The brokerage identification number is issued by SLA California.
2. Accept and/or reject the batch. An e-mail will be sent to the submission contact to confirm the acceptance or rejection of the batch. If the batch has been rejected, the message will contain a detailed description of the problem(s) and instructions to correct and resubmit the batch. The user must correct the batch and resubmit it through SLIP.

### 2.2.7. Table of XML Fields

The table below outlines the specific values for the elements in an XML schema prepared for the AMS Policy Data Submission Method. In the table, every element is individually addressed and a sample of the XML Structure is provided. The XML Structure column provides an example of the hierarchy and structural format that the submitted XML will be validated against. The Description column explains the element and the element's occurrence and length requirements.

XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<BatchDataSet ReportingState="CA" SchemaVersion="1" xmlns="http://ws.slacal.org/schema/slip_batch" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">	Root element of the XML file	1	1	-	-
<Batch>	Root element of the batch				
<Broker>	Broker details for the batch	1	1	-	-
<SLABrokerNumber>123456789</SLABrokerNumber>	A unique number assigned by SLA for each broker	1	1	0	9
<LicenseNumber>LIC12345</LicenseNumber>	Broker license number	1	1	0	9
<BrokerName>XYZ, Inc.</BrokerName>	Broker name	1	1	0	80
</Broker>	Ending Element				
<BrokerBatchNumber>100</BrokerBatchNumber>	Broker Batch Number	1	1	0	50
<BrokerBatchDate>2010-12-12</BrokerBatchDate>	Broker Batch Date	1	1	0	10
<BatchNotes>Notes for this batch.</BatchNotes>	Notes referring to the entire batch	0	1	0	4000
<LateFilingExplanation>Reason for late filing.</LateFilingExplanation>	Late filing explanation for the entire batch	0	1	0	4000
<Documents>	Documents associated with this batch. Within this element, you can choose to specify a folder name that will contain one or	0	1	-	-



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
	more documents or you can choose to specify filenames.				
<Document Xml_DocumentId="1">	Attached document details for the transaction (The Attribute "XML_DocumentId" must be unique within the submission)	1	unbound	-	-
<CustomDocumentId>1900101</CustomDocumentId>	Custom Document ID stored in the AMS system	1	1	0	50
<DocumentType>SL1</DocumentType>	Document type (DECLARATIONS PAGE, BINDER, ENDORSEMENT, CERTIFICATE, COVER NOTE, SL1, SL2, GAP, LINE SLIP, MULTIPLE, or OTHER)	1	1	0	50
<DocumentName>batchdocument.tiff</DocumentName>	Name of the document contained within the ZIP file accompanying this XML document.	1	1	0	200
</Document>	Ending Element				
Use either the 'Document' element above OR the 'Folder Name' element below:					
<FolderName>Batch_Docs</FolderName>	Name of a folder contained within the ZIP file containing documents that should be associated at the batch level	1	1	0	200
</Documents>	Ending Element				
<Policies>	Consists of all the policies submitted in the batch	1	unbound	-	-
<Policy Xml_PolicyId="1">	Policy details (The Attribute "XML_PolicyId" must be unique within the submission)	1	1	-	-
<Broker>	Broker details for the policy	1	1	-	-
<SLABrokerNumber>123456789</SLABrokerNumber>	A unique number assigned by SLA for each broker	1	1	0	9
<LicenseNumber>LIC12345</LicenseNumber>	Broker license number	1	1	0	9



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<BrokerName>XYZ, Inc.</BrokerName>	Broker name	1	1	0	80
</Broker>	Ending Element				
<CustomPolicyId>1001</CustomPolicyId>	Custom Policy ID stored in the AMS system	1	1	0	50
<PolicyNumber>A1234-1</PolicyNumber>	Policy number	1	1	0	50
<PreviousPolicyNumber>A1234-0</PreviousPolicyNumber>	For Renewals: previous policy number	1	1	0	50
<BinderNumber>12340001</BinderNumber>	Binder number	1	1	0	50
<EffectiveDate>2009-11-20</EffectiveDate>	For New, Renewal and eXtension endorsements: Policy effective date; For Binders: Binder effective date (YYYY-MM-DD)	1	1	0	10
<ExpirationDate>2010-11-20</ExpirationDate>	For New, Renewal and eXtension endorsements: Policy expiration date; For Binders: Binder expiration date (YYYY-MM-DD)	1	1	0	10
<GoodUntilCancelled>0</GoodUntilCancelled>	Is this policy good until cancelled?	1	1	0	1
<Insured>	Insured details for the policy	1	1	-	-
<Name>John Doe</Name>	Name of insured	1	1	0	150
<Address>	Address of insured	1	1	-	-
<AddressLine1>100 Main Road</AddressLine1>	Street Address line 1 of insured	1	1	0	50
<AddressLine2>Suite 200</AddressLine2>	Street Address line 2 of insured	1	1	0	50
<City>Beverly Hills</City>	City name of insured	1	1	0	30
<StateOrProvince>CA</StateOrProvince>	State or Province name of insured	1	1	0	50
<PostalCode>90210-4432</PostalCode>	Postal zip code of insured	1	1	0	10



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<Country>United States</Country>	Country name of insured	1	1	0	50
</Address>	Ending Element				
</Insured>	Ending Element				
<SecondaryInsuredName>Jane Doe</SecondaryInsuredName>	Secondary insured name	1	1	0	150
<IsMultipleStatePolicy>0</IsMultipleStatePolicy>	For New and Renewal: Is this a multi-state policy?	1	1	0	1
<IsMasterPolicy>1</IsMasterPolicy>	For New and Renewal: Is this a master policy?	1	1	0	1
<CaliforniaPercentAllocable>90</CaliforniaPercentAllocable>	For Multistate Policies with a policy effective date after the effective date of NRRA/AB315 specify the percent of the risk that is allocable to California	0	1	0	6
<ExemptCommercialPurchaser>0</ExemptCommercialPurchaser>	NRRA Exempt Commercial Purchaser also known as AB315 California Commercial Insured. This field is required for policies with an effective date after 7/21/2011 and should contain a boolean indicator.	0	1	0	1
<Transactions>	Consists of all the transactions for the policy	1	unbound	-	-
<Transaction Xml_TransactionId="1">	Transaction details for the policy (The Attribute "XML_TransactionId" must be unique within the submission)	1	1	-	-
<CustomTransactionId>1901</CustomTransactionId>	Custom Transaction ID stored in the AMS system	1	1	0	50
<TransactionType>R</TransactionType>	Transaction type (A:Audit Endorsement, C:Cancellation, E:Endorsement, N:New Business/Policy, O:Offset/Adjustment, R:Renewal, S:Security Update, X:Extension Endorsement)	1	1	1	50
<EndorsementEffectiveDate>2009-12-19</EndorsementEffectiveDate>	For Endorsement: Endorsement effective date; For Cancellations: Cancellation date (YYYY-MM-DD)	1	1	0	10



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<EndorsementNumber>9876</EndorsementNumber>	For Endorsements: Endorsement number	1	1	0	10
<SICCode>Amusement & Recreation Services</SICCode>	SIC code. SLIP accepts either the numeric value or the text of the code. Empty values are also acceptable.	1	1	0	150
Use either the 'SIC Code' element above OR the 'NAICS Code' element below					
<NAICSCode>Amusement</NAICSCode>	NAICS code	1	1	0	150
<Notes>string</Notes>	Notes for the transaction	1	1	0	unbound
<LateFilingExplanation>string</LateFilingExplanation>	Late filing explanation for policies filed 60 days after the policy effective date	1	1	0	unbound
<Premium>2231.45</Premium>	Amount of premium for the transaction	1	1	0	12
<InvoiceDate>2009-12-11</InvoiceDate>	Invoice date of the transaction (YYYY-MM-DD)	1	1	0	10
<Coverages>	Consists of all the coverages for the transaction	0	1	-	-
<Coverage>	Coverage details for the transaction	1	unbound	-	-
<CoverageType>Miscellaneous</CoverageType>	Coverage type	0	1	0	150
<CoverageDescription>Accident</CoverageDescription>	Coverage description	1	1	0	150
<CoverageCode>700</CoverageCode>	Coverage code	1	1	0	-
<LayerGroup>	Layer group containing layering and layer percentage	0	1	-	-
<LayerDescription>Layer 1</LayerDescription>	Layer description	1	1	0	75
<LayerData>	Layer detail record containing insurance company or layering of insurance companies and the percentage allocated to the layer	0	unbound	-	-





XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<Insurer>	Insurer details for the coverage	1	1	-	-
<Name>First Insurance</Name>	Name of insurer	1	1	0	150
<NAICNumber>A-10349</NAICNumber>	NAIC number assigned to the insurer	1	1	0	16
</Insurer>	Ending Element				
<LayerPercentage>100.00000</LayerPercentage>	Percentage allocated to the insurance company or layering of insurance companies	1	1	0	8
</LayerData>	Ending Element				
</LayerGroup>	Ending Element				
Use either the 'Layer Group' element above OR the 'Insurer' element below					
<Insurer>	Insurer details for the coverage	1	1	-	-
<Name>First Insurance</Name>	Name of insurer	1	1	0	150
<NAICNumber>A-10349</NAICNumber>	NAIC number assigned to the insurer	1	1	0	16
</Insurer>	Ending Element				
<CoveragePremium>2231.45</CoveragePremium>	Amount of the coverage premium	1	1	0	12
</Coverage>	Ending Element				
</Coverages>	Ending Element				
<Fees>	Consists of all the fees for the transaction	0	unbound	-	-
<Fee>	Fee details for the transaction	1	1	-	-



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<FeeType> <b>Broker Fee</b> </FeeType>	Fee type (Policy fee, Inspection fee, Broker fee, or Other)	1	1	0	50
<FeeDescription> <b>string</b> </FeeDescription>	Fee description	1	1	0	150
<IncludeInPremium> <b>1</b> </IncludeInPremium>	Indicates whether or not the fee is included in the total item premium amount	1	1	0	1
<FeeAmount> <b>198.43</b> </FeeAmount>	Fee amount	1	1	0	12
</Fee>	Ending Element				
</Fees>	Ending Element				
<StateAllocations>	Consists of all the state allocations for the transaction	0	unbound	-	-
<StateAllocationMethod> <b>Premium</b> </StateAllocationMethod>	Method of allocation used CIC 1775.5	1	1	0	50
<StateAllocation>	State allocation details for the transaction	1	1	-	-
<StateName> <b>CA</b> </StateName>	Name of the state allocated	1	1	0	50
<StatePremium> <b>2231.45</b> </StatePremium>	Amount of premium allocated to the state	1	1	0	12
</StateAllocation>	Ending Element				
<b>Use either the 'Premium' method above OR the 'Percentage' method below</b>					
<StateAllocationMethod> <b>Percentage</b> </StateAllocationMethod>	Method of allocation used CIC 1775.5	1	1	0	50
<StateAllocation>	State allocation details for the transaction	1	1	-	-
<StateName> <b>CA</b> </StateName>	Name of the state allocated	1	1	0	50
<StatePercentage> <b>99.99999</b> </StatePercentage>	Percentage of premium allocated to the state	1	1	0	8



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
</StateAllocation>	Ending Element				
<TotalPremiumAllStates>1000.00 </TotalPremiumAllStates>	Total amount of premium for all the states under this policy	0	1	0	12
</StateAllocations>	Ending Element				
<Certificates>	Consists of all the certificates for the transaction	0	unbound	-	-
<Certificate>	Certificate details for the transaction	1	1	-	-
<CertificateNumber>A-00001</CertificateNumber>	Certificate number for Master Policy	1	1	0	50
<Insured>	Insured details for the certificate	1	1	-	-
<Name>John Doe</Name>	Certificate insured name	1	1	0	150
<Address>	Address of insured	1	1	-	-
<AddressLine1>100 Main Road</AddressLine1>	Street Address line 1 of insured	1	1	0	50
<AddressLine2>Suite 200</AddressLine2>	Street Address line 2 of insured	1	1	0	50
<City>Beverly Hills</City>	City name of insured	1	1	0	30
<StateOrProvince>CA</StateOrProvince>	State or Province name of insured	1	1	0	50
<PostalCode>90210-5633</PostalCode>	Postal zip code of insured	1	1	0	10
<Country>United States</Country>	Country name of insured	1	1	0	50
</Address>	Ending Element				
</Insured>	Ending Element				
<CertificateEffectiveDate>2010-12-25</CertificateEffectiveDate>	Certificate effective date	1	1	0	10



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<b>&lt;CertificateExpirationDate&gt;2011-12-25&lt;/CertificateExpirationDate&gt;</b>	Certificate expiration date	1	1	0	10
<b>&lt;CertificatePremium&gt;1020.25&lt;/CertificatePremium&gt;</b>	Certificate premium	1	1	0	12
<b>&lt;/Certificate&gt;</b>	Ending Element				
<b>&lt;/Certificates&gt;</b>	Ending Element				
<b>&lt;SL1&gt;</b>	SL1 form details for the transaction	0	1	-	-
<b>&lt;PolicyNumber&gt;A1-123-1&lt;/PolicyNumber&gt;</b>	Policy number	1	1	0	50
<b>&lt;CaliforniaPremium&gt;1020.25&lt;/ CaliforniaPremium &gt;</b>	California Premium	1	1	0	12
<b>&lt;CoverageOnExportList&gt;Yes, Coverage on Export List &lt;/CoverageOnExportList&gt;</b>	The submission is either for an insurance Coverage or risk on Export list and diligent search report	0	1	0	150
<b>Use either the 'Coverage on Export List' element above OR the 'Diligent Search Report' element below.</b>					
<b>&lt;DiligentSearchReport&gt;Yes, Diligentsearch performed, see attached SL2 Form&lt;/DiligentSearchReport&gt;</b>	The submission is for either an insurance Coverage or risk on Export list and diligent search report.	0	1	0	150
<b>&lt;BrokerName&gt;Bob Broker&lt;/Brokeraame&gt;</b>	Name of Broker	1	1	0	150
<b>&lt;BrokerLicenseNumber&gt;L123456 Dunn&lt;/BrokeLicenseNumber&gt;</b>		0	1	0	9
<b>Use either the 'BrokerLicenseNumber' element above OR the 'Organization' element below.</b>					
<b>&lt;Organization&gt;</b>	Organization details for the SL1 form	1	1	-	-
<b>&lt;Name&gt;ABC, Inc.&lt;/Name&gt;</b>	Name of Organization	1	1	0	150
<b>&lt;LicenseNumber&gt;LIC12345&lt;/LicenseNumber&gt;</b>	License number of Organization	1	1	0	50



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
</Organization>	Ending Element				
<Insured>	Insured details for the SL1 form	1	1	-	-
<Name>John Doe</Name>	Name of insured	1	1	0	150
<Address>	Address of insured	1	1	-	-
<AddressLine1>100 Main Road</AddressLine1>	Street Address line 1 of insured	1	1	0	50
<AddressLine2>Suite 200</AddressLine2>	Street Address line 2 of insured	1	1	0	50
<City>Beverly Hills</City>	City name of insured	1	1	0	30
<StateOrProvince>CA</StateOrProvince>	State or Province name of insured	1	1	0	50
<PostalCode>90210-5633</PostalCode>	Postal zip code of insured	1	1	0	10
<Country>United States</Country>	Country name of insured	1	1	0	50
</Address>	Ending Element				
</Insured>	Ending Element				
<RiskDescription>string</RiskDescription>	Description of risk	1	1	0	150
<RiskLocation>	Risk location details	1	1	-	-
<AddressLine1>200 Main Street</AddressLine1>	Street Address line 1 of risk location	1	1	0	50
<AddressLine2>Suite 100</AddressLine2>	Street Address line 2 of risk location	1	1	0	50
<City>Beverly Hills</City>	City name of risk location	1	1	0	30
<StateOrProvince>CA</StateOrProvince>	State or Province name of risk location	1	1	0	50
<PostalCode>90210-5677</PostalCode>	Postal zip code of risk location	1	1	0	10



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<Country>United States</Country>	Country name of risk location	1	1	0	50
</RiskLocation>	Ending Element				
<ExportOrCoverageCodes>	Consists of all the Export or Coverage codes for the SL1 form	0	unbound	-	-
<ExportOrCoverageCode>	the Export or Coverage code details for the SL1 form	1	1	-	-
<CoverageCode>Accident</CoverageCode>	Export or Coverage code	1	1	0	150
<CoverageDescription>string</CoverageDescription>	Export or Coverage description	1	1	0	150
</ExportOrCoverageCode>	Ending Element				
</ExportOrCoverageCodes>	Ending Element				
<NonAdmittedInsurers>	Consists of all the SL1 Non-Admitted Insurer	0	unbound	-	-
<SL1NonAdmittedInsurer>	SL1 Non-Admitted Insurer details for the transaction	1	1	-	-
<NonAdmittedInsurerName>NAI, Inc. </NonAdmittedInsurerName>	Name of non-admitted insurer	1	1	0	150
<NAICNumber>7777</NAICNumber>	NAIC number of non-admitted insurer	1	1	0	16
<PremiumPercentage>100.00000</PremiumPercentage>	Non-admitted insurer's premium percentage	1	1	0	8
</SL1NonAdmittedInsurer>	Ending Element				
</NonAdmittedInsurers>	Ending Element				
<DateSigned>2010-09-12</DateSigned>	Date signed	1	1	0	10
</SL1>	Ending Element				



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<GAP>	GAP form details for the transaction	0	1	-	-
<PolicyNumber> <b>A1-123-1</b> </PolicyNumber>	Policy number	1	1	0	50
<Layers>	Consists of all the GAP layers	1	unbound	-	-
<GAPLayer>	GAP layer details	1	1	-	-
<LayerNumber> <b>1</b> </LayerNumber>	Layer number (Primary policy is layer number 1)	1	1	0	1
<LimitOfLiability> <b>1525.00</b> </LimitOfLiability>	Limit of liability	1	1	0	12
<ExcessOfUnderlyingLimits> <b>500.50</b> </ExcessOfUnderlyingLimits>	Excess of underlying limits	1	1	0	12
<GapInsurersPercentage> <b>100.00000</b> </GapInsurersPercentage>	Percentage of layer with GAP Insurers	1	1	0	8
</GAPLayer>	Ending Element				
</Layers>	Ending Element				
<TotalNumberOfLayers> <b>2</b> </TotalNumberOfLayers>	Total number of Layers	1	1	0	3
<TotalLimitsOfLiability> <b>2231.45</b> </TotalLimitsOfLiability>	Total limits of Liability for all layers combined	1	1	0	12
<TotalPercentageOfGapInsurers> <b>90.00000</b> </TotalPercentageOfGapInsurers>	Total percentage of GAP Insurers for all layers combined	1	1	0	8
<AssociatedLayerNumber> <b>1</b> </AssociatedLayerNumber>	This submission is for layer number (Layer number associated with the submission)	1	1	0	1
<GapInsurers>	Consists of all the GAP insurers	0	unbound	-	-
<GapInsurer>	GAP insurer details	1	1	-	-



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<Name>ABC, Inc.</ Name>	Name of insurer	1	1	0	150
<NAICNumber>55555</NAICNumber>	NAIC number assigned to the insurer	1	1	0	16
<ParticipationPercentage>50.00000 </ParticipationPercentage>	GAP Insurer participating percentage for this layer or policy	1	1	0	8
</GapInsurer>	Ending Element				
</GapInsurers>	Ending Element				
<DateSigned>2010-09-12</DateSigned>	Date signed	1	1	0	10
</GAP>	Ending Element				
<SL2>	SL2 form details for the transaction	0	1	-	-
<CoverageOnExportList>Yes, Coverage on Export List </CoverageOnExportList>	Choice between the submission is for an insurance Coverage or risk on Export list and diligent search report	0	1	0	150
Use either the 'Coverage on Export List' element above OR the 'Diligent Search Report' element below.					
<DiligentSearchReport>Yes, Diligentsearch performed, see attached SL2 Form</DiligentSearchReport>	Choice between the submission is for an insurance Coverage or risk on Export list and diligent search report.	0	1	0	150
<BrokerName>Bob Broker</BrokerName>	Broker name for the SL2 form	1	1	0	150
<BrokerLicenseNumber>LIC9999 </BrokerLicenseNumber>	License number of the Broker	1	1	0	50
Use either the 'BrokerLicenseNumber' element above OR the 'Organization' element below.					
<Organization>	Organization details for the SL1 form	1	1	-	-
<Name>ABC, Inc.</Name>	Name of Organization	1	1	0	150





XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<LicenseNumber>LIC12345</LicenseNumber>	License number of Organization	1	1	0	50
</Organization>	Ending Element				
<Insured>	Insured details for the SL2 form	1	1	-	-
<Name>John Doe</Name>	Name of the insured	1	1	0	150
<Address>	Address of insured	1	1	-	-
<AddressLine1>100 Main Road</AddressLine1>	Street Address line 1 of insured	1	1	0	50
<AddressLine2>Suite 200</AddressLine2>	Street Address line 2 of insured	1	1	0	50
<City>Beverly Hills</City>	City name of insured	1	1	0	30
<StateOrProvince>CA</StateOrProvince>	State or Province name of insured	1	1	0	50
<PostalCode>90210-5633</PostalCode>	Postal zip code of insured	1	1	0	10
<Country>United States</Country>	Country name of insured	1	1	0	50
</Address>	Ending Element				
</Insured>	Ending Element				
<RiskDescription>string</RiskDescription>	Description of risk	1	1	0	200
<RiskLocation>	Address of risk	1	1	-	-
<AddressLine1>200 Main Street</AddressLine1>	Street Address line 1 of risk location	1	1	0	50
<AddressLine2></AddressLine2>	Street Address line 2 of risk location	1	1	0	50
<City>Beverly Hills</City>	City name of risk location	1	1	0	30
<StateOrProvince>CA</StateOrProvince>	State or Province name of risk location	1	1	0	50



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<PostalCode>90210-5633</PostalCode>	Postal zip code of risk location	1	1	0	10
<Country>United States</Country>	Country name of risk location	1	1	0	50
</RiskLocation>	Ending Element				
<ExportOrCoverageCodes>	Consists of all the Export or Coverage codes for the SL2 form	0	unbound	-	-
<ExportOrCoverageCode>	the Export or Coverage code details for the SL2 form	1	1	-	-
<CoverageCode>700</CoverageCode>	Export or Coverage code	1	1	0	-
<CoverageDescription> Accident </CoverageDescription>	Export or Coverage description	1	1	0	150
</ExportOrCoverageCode>	Ending Element				
</ExportOrCoverageCodes>	Ending Element				
<IsGoodDriver>1</IsGoodDriver>	Indicates whether or not the insured is a good driver	1	1	0	1
<IncludedCAARP>1</IncludedCAARP>	Indicates whether or not CAARP coverage is included	1	1	0	1
<CAARPineligible>0</CAARPineligible>	Indicates whether or not the risk is CAARP ineligible	1	1	0	1
<IsSmallEmployer>1</IsSmallEmployer>	Indicates whether or not the insured is a small employer	1	1	0	1
<PurchasingGroupName>Cal Association</PurchasingGroupName>	Name of the Risk Purchasing Group	1	1	0	150
<PurchasingGroupAddress>1100 King Drive, Los Angeles, CA 90575</PurchasingGroupAddress>	Address of the Risk Purchasing Group	1	1	0	150
<DiligentEfforts>string</DiligentEfforts>	Description of the diligent efforts made to place this coverage with admitted insurers and how the search was performed	1	1	0	unbound
<NameOfSearchPerformer>Jane Doe</NameOfSearchPerformer>	Full name of person who performed diligent search	1	1	0	4000



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<WasSubmittedToAdmittedInsurers>1</WasSubmittedToAdmittedInsurers>	Indicates whether or not the risk has been submitted to at least three admitted insurers	1	1	0	1
<AdmittedInsurers>	Consists of all the SL2 Admitted Insurer	0	unbound	-	-
<SL2AdmittedInsurer>	SL2 Admitted Insurer details for the transaction	1	1	-	-
<CompanyName>QWE, Inc.</CompanyName>	Name of admitted insurance company	1	1	0	150
<NAICNumber>876543</NAICNumber>	NAIC number assigned to the admitted insurance company	1	1	0	16
<RepresentativeName>David King</RepresentativeName>	First and last name of the admitted insurance company representative	1	1	0	150
<RepresentativePhone>9008007000</RepresentativePhone>	Phone number of the admitted insurance company representative	1	1	0	50
<DeclinationWebsite>www.decline.org</DeclinationWebsite>	Online Declination website	1	1	0	200
<RepresentativeType>Agent</RepresentativeType>	Representative Type (Employee or Agent)	1	1	0	50
<DeclinationMonth>2</DeclinationMonth>	Month of Declination	1	1	0	2
<DeclinationYear>2009</DeclinationYear>	Year of Declination	1	1	0	4
<DeclinationCode>Refused To State</DeclinationCode>	Declination code (Capacity Reached, Underwriting Reason, Refused To State, or Other)	1	1	0	50
</SL2AdmittedInsurer>	Ending Element				
</AdmittedInsurers>	Ending Element				
<DeterminedLessThan3Insurers>1</DeterminedLessThan3Insurers>	Indicates whether or not the risk has been submitted to fewer than three admitted insurers	1	1	0	1



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
<b>&lt;ReasonLessThan3Insurers&gt;string &lt;/ReasonLessThan3Insurers&gt;</b>	Explanation of why the risk was submitted to fewer than three admitted insurers	1	1	0	unbound
<b>&lt;DeterminationDescription&gt;string &lt;/DeterminationDescription&gt;</b>	Description of how the determination made	1	1	0	unbound
<b>&lt;DateSigned&gt;2010-09-12&lt;/DateSigned&gt;</b>	Date signed	1	1	0	10
<b>&lt;/SL2&gt;</b>	Ending Element				
<b>&lt;Documents&gt;</b>	Documents associated with this transaction. Within this element, you can choose to specify a folder name that will contain one or more documents, or you can choose to specify filenames.	0	1	-	-
<b>&lt;Document Xml_DocumentId="2"&gt;</b>	Attached document details for the transaction (The Attribute "XML_DocumentId" must be unique within the submission)	1	unbound	-	-
<b>&lt;CustomDocumentId&gt;1900101&lt;/CustomDocumentId&gt;</b>	Custom Document ID stored in the AMS system	1	1	0	50
<b>&lt;DocumentType&gt;SL1&lt;/DocumentType&gt;</b>	Document type (DECLARATIONS PAGE, BINDER, ENDORSEMENT, CERTIFICATE, COVER NOTE, SL1, SL2, GAP, LINE SLIP, MULTIPLE, or OTHER)	1	1	0	50
<b>&lt;DocumentName&gt;SL1_A1-123-1.tiff&lt;/DocumentName&gt;</b>	Name of the document contained within the ZIP file accompanying this XML document.	1	1	0	200
<b>&lt;/Document&gt;</b>	Ending Element				
<b>Use either the 'Document' element above OR the 'Folder' element below.</b>					
<b>&lt;FolderName&gt;folder name&lt;/FolderName&gt;</b>	Name of a folder contained within the ZIP file containing documents that should be associated at the batch level	1	1	0	200
<b>&lt;/Documents&gt;</b>	Ending Element				



XML Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
</Transaction>	Ending Element				
</Transactions>	Ending Element				
</Policy>	Ending Element				
</Policies>	Ending Element				
</Batch>	Ending Element				
</BatchDataSet>	Ending Element				

### 2.3. Additional XML Information

XML creation software may help you examine and work within the parameters of the XML schema. These tools include Liquid XML Studio, Stylus XML Studio, XML Spy, and others. XML creation software will also validate your file prior to submission.

The following websites contain valuable information regarding the XML Standard and the UCC XML Standard as well as some information concerning XML tools.

- <http://www.w3.org/XML>
- <http://www.xml.org>
- <http://www.xml.com>
- <http://www.w3schools.com/xml/default.asp>
- <http://www.w3schools.com/Schema/default.asp>

### 3. Bulk Upload (XML)

#### 3.1. Description

The Bulk Upload method allows brokerages to submit policy and transaction data for multiple policies at once. This process will especially benefit brokerages that file a large amount of policy data with SLA California since a single XML file may contain information for multiple policies.

Brokerages that store data in a centralized data management system can make use of the Bulk Upload method. The following list provides a high-level list of the steps contained within the Bulk Upload process:

1. The brokerage will generate an XML file containing the policy data it wishes to submit to SLA California. Typically, the XML file will include policy data that was added or modified within a specified date range or since the last XML batch submission.
2. The brokerage should also include any associated policy documentation (PDF, PDF/A, PNG, TIFF, and JPG/JPEG) that is associated with the policy data contained within the XML file. The policy documents will be referenced by filename within the XML file.
3. The brokerage will create a ZIP file containing the XML file and the associated policy documentation (again, referenced by filename within the XML file).
4. The brokerage will log in to SLIP to upload the ZIP file containing the XML file and all associated documents.

#### 3.2. Prerequisites

Brokerages may submit policy data using the Bulk Upload method if the following requirements are met: A SLIP user account must exist for a submitting brokerage. SLA California will create one master user account for each participating brokerage. The master user within each brokerage will have the ability to create other user accounts for that brokerage.

A supported web browser must be used. The following web browsers will be supported:

Operating System	Web Browser	Version
Microsoft Windows	Microsoft Internet Explorer	8+
	Mozilla FireFox	3+
	Apple Safari	4.0.3+
	Google Chrome	4.1+
MAC OS	Apple Safari	4+
	Mozilla FireFox	3+

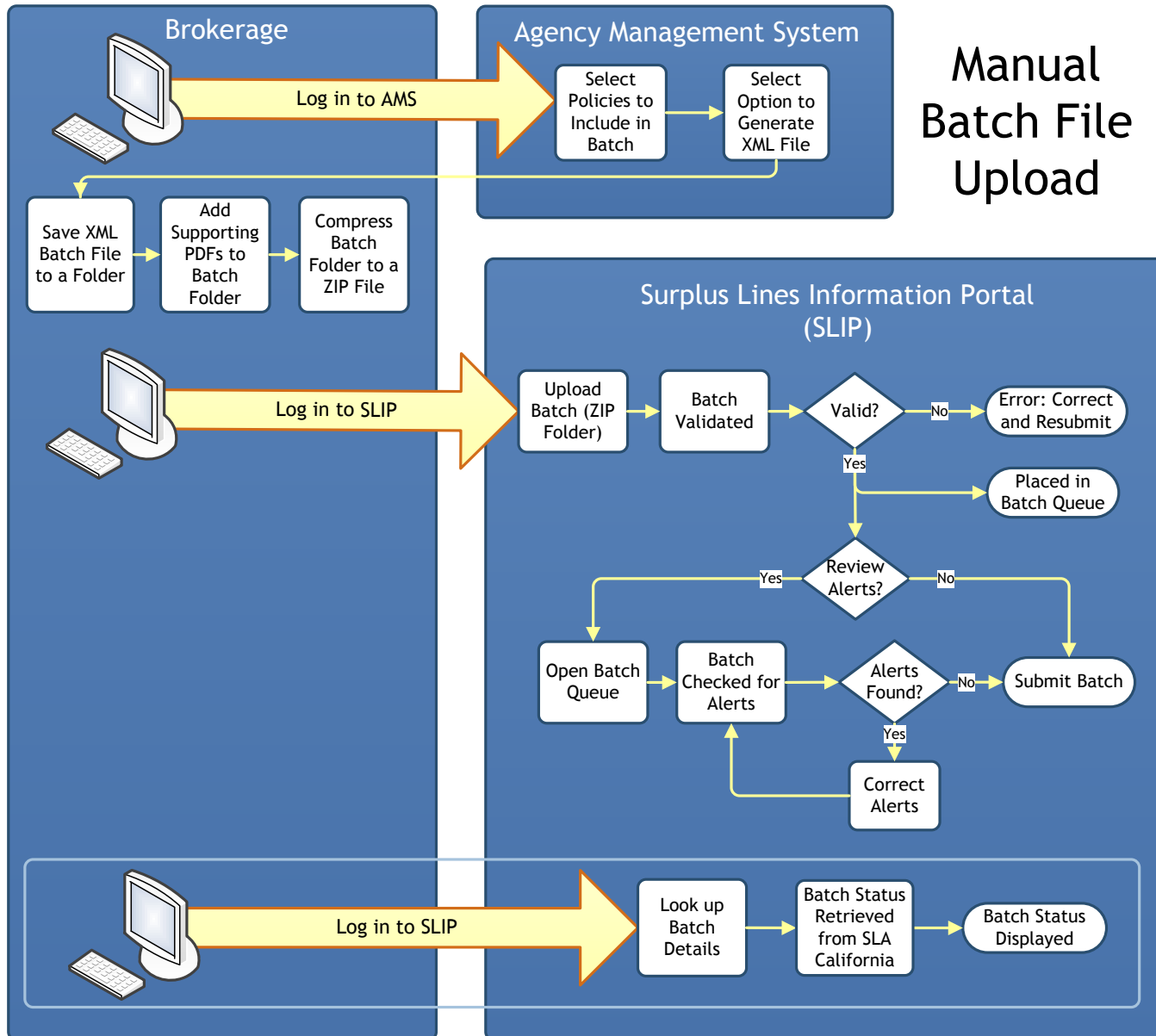
An SLA California Broker Number (SLA #) is required within the XML file for the submitting brokerage. To identify your SLA #, please contact SLA California.

### 3.3. Process

This section identifies the steps required to create and submit policy information using the manual batch file upload process. The graphic below is a high-level representation of the process flow and indicates the systems involved.



# Manual Batch File Upload



### **3.3.1. Create Batch File**

The first step in the Bulk Upload process is to create the XML batch file with the policy data. Refer to section 3.1.6 of this document for details about creating the XML batch file.

For the Bulk Upload process, the brokerage management system may be configured to create the entire batch with all policy documentation already included in a ZIP file. If it is not configured this way, additional steps may be taken by users to manually add the desired policy documents to either the root of the ZIP file or a folder referenced within the batch ZIP file. See section 2.2.1. Batch File Size and Structure for more details about files and folders within the batch ZIP file.

### **3.3.2. Log in to SLIP**

Using a supported web browser, go to the SLA California SLIP website. Enter your username and password. This will establish a secure connection and validate your identity.

SLA California will create one administrative user for each brokerage. The administrative user for each brokerage will have the ability to create other users for that brokerage.

### **3.3.3. Upload and Submit the File**

Go to the Batch Submission page in SLIP. Following the instructions on this page, browse to and select the compressed ZIP file containing the XML file and associated policy documents (PDF, PDF/A, PNG, TIFF, and JPG/JPEG). Submit the file for upload.

SLIP will assign a Batch Source for all batches. Batches that are uploaded manually will have "XML" listed as the batch source.

## Submission Type Selection

Please select a submission type from the options below:

☒ Create Batch

Click "Create Batch" to create a batch with policies you have already created in SLIP.

☐ Bulk Upload (Advanced: XML Format Only)

Click "Bulk Upload" to upload policy data in bulk into SLIP. The Bulk Upload feature requires a ZIP archive containing the bulk policy data in a single XML file along with any additional related documents. You will have the ability to examine these policies and check for errors before submitting them to the SLA.

For details on the required XML format, [review the Bulk Upload procedures](#).

☐ Upload Batch Image File (Broker Extranet Method)

Select this option to upload a group of document images into SLIP for submission to the SLA. This is similar to the Broker Extranet process. This method does not have the benefit of "early warning" alerts that are used for user review prior to submission, but are still reviewed, processed and filed by the Data Analysis Department at the SLA.

\*Please note that you will only get a single email notification when your batch is submitted to the SLA. There will not be any batch status updates for this option.

Next >

Figure 1 - Create New XML Bulk Upload Batch



## Create New Batch Wizard

SLIP >> File Selection (Step 2 of 2)

### File Selection

Select the ZIP XML file for Bulk Upload

[Browse...](#)

☐ Ignore alerts during submission

[< Back](#)[Save](#)[Submit](#)

**Figure 2 - Select ZIP XML File for Bulk Upload**



Your batch (SLIP Batch ID 12018) has been successfully saved but not submitted. Once your batch is finished processing you may review the batch on the View Batch screen.



## Create New Batch Wizard

SLIP >> File Selection (Step 2 of 2)

### File Selection

Select the ZIP XML file for Bulk Upload

[Browse...](#)

☐ Ignore alerts during submission

[< Back](#)[Save](#)[Submit](#)

**Figure 3 - Bulk Upload ZIP XML File Saved**

### 3.3.4. SLIP Validates the File

After you successfully upload a batch file in SLIP, the system will queue the submission for processing. When the system is ready to process the submission, the validation process will begin.

The first step is to validate the format and structure of the XML file as identified in the XML Schema. This step must be successfully completed before any additional processing can take place. If the initial validation process is successful, the XML file will continue to the secondary validation process. The secondary validation process validates the policy data contained within the XML file itself. If either validation process is unsuccessful, the file will be rejected. The XML file format and/or data will have to be corrected and resubmitted.

Whether the file is accepted or rejected, an e-mail will be sent to the user or users associated with the SLA broker number. If the submission was successful, the email

will include the filing number and filing date. If this submission was rejected, the email will contain the date and time the file import was attempted and the reason(s) the file was rejected. In both scenarios, the Batch Submission page within SLIP will display the processing status of any submission.

### 3.3.5. Monitor the Batch Submission Status

After confirming that your batch file was successfully uploaded in SLIP, you may monitor the batch progress in the SLIP Batch Submission page. The page will contain the date the file was submitted and received by SLA California. Rejected submissions should be corrected and resubmitted in a timely manner. The following table defines the batch statuses.

External Status	Description
<b>UPLOADED</b>	SLIP has identified and received a batch. The received batch will upload into SLIP automatically and be validated.
<b>INVALID UPLOAD</b>	The batch has failed validation or was not properly imported into SLIP. SLA California has not accepted a batch.
<b>PENDING USER ACTION</b>	The batch is currently Saved in SLIP and requires user action before the batch will upload to SLA.
<b>PENDING SLA REVIEW</b>	The batch was successfully uploaded to SLA California and is pending SLA Review.
<b>CLOSED BY SLA</b>	The Batch was invoiced by the SLA. <i>Note: Closed batches can contain Open Tags.</i>
<b>RETURNED BY SLA</b>	The batch either was received and returned by the SLA or was not successfully uploaded. The SLA will notify users in either instance by both email and written letter.

### 3.3.6. Batch File is Accepted or Rejected

If the file has been accepted for import, no further action is required. As mentioned in section 3.2.5., you may monitor the batch import process on the Batch Submission page.

If the file has been rejected for import, you must correct the issue(s) identified in the rejection email and resubmit the batch file. If you have questions regarding batch file rejection or resubmission, please contact SLA California.

## 4. API Batch Submission (XML)

### 4.1. Description

The API structure will provide the means for third party applications to submit insurance policy data and documentation, on behalf of a broker or brokerage, to SLA California. It also provides the means for the AMS to receive feedback in regards to the acceptance of the submitted data by the SLA.

All endpoints will be based upon the hypertext transfer protocol (HTTP) and will use the simple object access protocol (SOAP) to exchange structured data sets as defined in this document.

### 4.2. Pre-requisites

Brokerages may submit policy data using the API Batch Submission Method if the following requirements are met:

For a brokerage management system to be allowed to interact with the API on behalf of a user, it must first collect a set of credentials from the user. This set of credentials will include a SLA broker number, username, and API key (user token) value. Users should be able to obtain these values from their SLIP user profile page. The username and key pair uniquely identify a user and indicate that users have granted the brokerage management system permission to perform tasks on their behalf. See section 4.4.1., Credential Verification Endpoint for the specific method used to verify the credentials.

**My Account**  
SLIP >> My Account

[< Back](#)

### My Profile

Profile	Permissions
User Name:	<input checked="" type="checkbox"/> Create/Edit Policies
Broker License Number: 12345	<input checked="" type="checkbox"/> View Tags
First Name:	<input checked="" type="checkbox"/> Respond to Tags
Last Name:	<input checked="" type="checkbox"/> Create/Edit Batches
Phone Number:	<input checked="" type="checkbox"/> Submit Batches
Phone Extension:	<input checked="" type="checkbox"/> Submit Batches With Alerts
Email Address:	<input checked="" type="checkbox"/> Submit XML/AMS Batches without Previewing Data in SLIP
Date Last Accessed:	<input checked="" type="checkbox"/> View Invoices (logged in Location only)
<a href="#">Change Password</a>	<input checked="" type="checkbox"/> Pay Invoices (logged in Location only)
AMS Token: 1ccbd571-40ae-4c54-8e0a-9db20b667cda	<input checked="" type="checkbox"/> View and Pay Invoices for all Associated Locations
<a href="#">Regenerate</a>	<input checked="" type="checkbox"/> View Reports
	<input checked="" type="checkbox"/> View My Account Information
	<input checked="" type="checkbox"/> Maintain My Profile and Email Notification Settings
	<input checked="" type="checkbox"/> Maintain Insurers
	<input checked="" type="checkbox"/> Maintain Transactors
	<input checked="" type="checkbox"/> Maintain User Accounts

Figure 4 - AMS Token

The SLA broker number, username and API key values must be supplied within the SOAP header with every request to the API. It is recommended that the brokerage management system invoke the credential verification endpoint prior to submitting data to ensure that the credentials are valid.

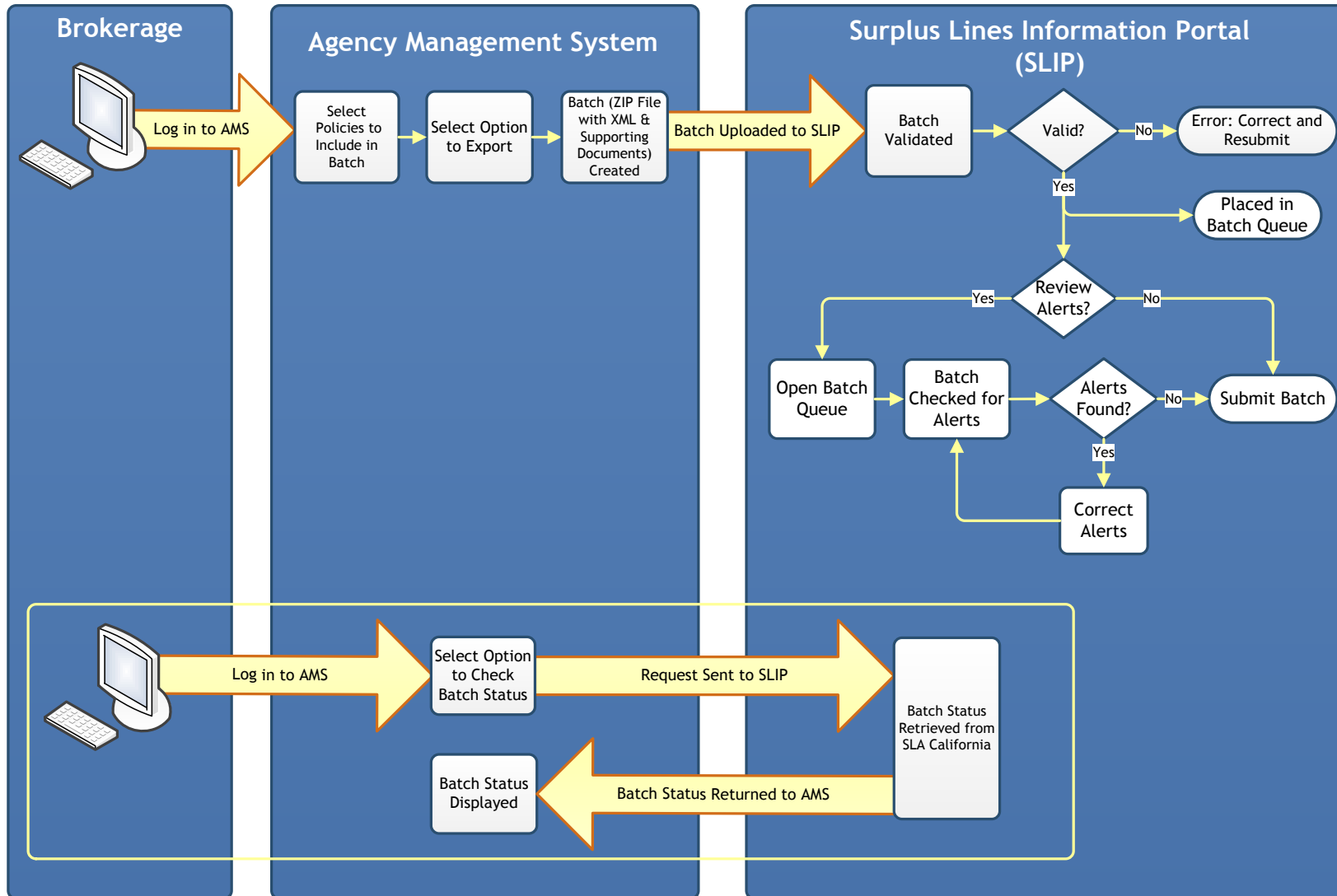
A SLIP user account must exist for a submitting brokerage. SLA California will create one administrative user account for each participating brokerage. The administrative user in each brokerage will have the ability to create other user accounts for that brokerage.

An SLA California Broker Number (SLA #) is required within the XML file for the submitting brokerage. To identify your SLA #, please contact SLA California.

### **4.3. Process**

This section identifies the steps required to create and submit policy information using the API batch submission process. The high-level representation of the process flow below indicates the systems involved.

## API Batch Submission





#### **4.3.1. Create Batch File**

The first step in the API batch submission process is to create the XML batch file with the policy data. Refer to section 2.2.4. of this document for details about creating the XML batch file. For the API batch submission process, the brokerage management system will create the batch file automatically.

#### **4.3.2. Submit Batch File**

Users will select the option in their brokerage management system to submit the batch information to SLIP. See section 5.4.2., Upload Batch Filing Endpoint for the specific method used to upload the file.

SLIP will assign a Batch Source for all batches. Batches uploaded by the API method will have "AMS" listed as the batch source.

#### **4.3.3. SLIP Validates the File**

After you successfully upload a batch in SLIP, the system will queue the submission for processing. When the system is ready to process the submission, the validation process will begin.

The first step is to validate the format and structure of the batch as identified in the XML Schema. This step must be successfully completed before any additional processing can take place. If the initial validation process is successful, the batch will continue to the secondary validation process. The secondary validation process validates the policy data contained within the XML file itself. If either validation process is unsuccessful, the batch will be rejected. The XML file format and/or data will have to be corrected and resubmitted.

Whether the file is accepted or rejected, an e-mail will be sent to the submission contact identified within the batch. If the submission was successful, the email will include the filing number and filing date. If this submission was rejected, the email will contain the date and time the file import was attempted and the reason(s) the file was rejected. In both scenarios, the Batch Submission page within SLIP will display the processing status of any submission. The user may also use the Check Status Endpoint to get the status of the batch.

#### **4.3.4. Monitor the Batch Submission Status**

After confirming that your batch was successfully uploaded in SLIP, you may monitor the batch progress in the SLIP Batch Submission page, or use the Check Status Endpoint method. The page will contain the date the batch was submitted and received by SLA California. Rejected submissions should be corrected and resubmitted in a timely manner. The following table defines the batch statuses.

See section 4.4.3. Check Status Endpoint for the specific method used to monitor the batch status.

External Status	Description
<b>UPLOADED</b>	SLIP has identified and received a batch. The received batch will upload into SLIP automatically and validated.
<b>INVALID UPLOAD</b>	The batch has failed validation or was not properly imported into SLIP. SLA California has not accepted a batch.
<b>PENDING USER ACTION</b>	The batch is currently Saved in SLIP and requires user action before the batch will Upload to SLA.
<b>PENDING SLA REVIEW</b>	The batch was successfully uploaded to SLA California and is pending SLA Review.
<b>CLOSED BY SLA</b>	The Batch was invoiced by the SLA. <i>Note: Closed batches can contain Open Tags.</i>
<b>RETURNED BY SLA</b>	The batch either was received and returned by the SLA or was not successfully uploaded. The SLA will notify users in either instance by both email and written letter.

#### 4.3.5. Review Notifications

The user will have the option to review the notifications generated by the validation of the file. If the user has the appropriate permissions, they may choose to submit the batch without correcting any alerts noted; however, some users will not have this privilege and must correct the alerts before the batch may be submitted to SLA California. See section 5.4.5. Get Notifications Endpoint for the specific method used to retrieve the notifications.

#### 4.4. Methods

This section contains the specific methods used in the API batch submission method. Each method is referenced in a step of the API batch submission process, above.

#### **4.4.1. Credential Verification Endpoint**

Upon collecting API credentials from the user, it is recommended that the AMS invoke this endpoint to verify access to the API on the user's behalf. This will ensure that the user's account is active and verify the user's identity.

It is also recommended that the AMS verify the user's credentials prior to each data submission to ensure that the credentials remain valid.

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

**Request message:**

```
POST /AMSBatchFiling.asmx HTTP/1.1
Host: serverhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://test.ws.slacal.org/ws-slip/VerifyCredentials"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <AuthenticationHeader xmlns="http://test.ws.slacal.org/ws-slip">
      <SLABrokerNumber>string</SLABrokerNumber>
      <UserName>string</UserName>
      <APIKey>string</APIKey>
    </AuthenticationHeader>
  </soap:Header>
  <soap:Body>
    <VerifyCredentials xmlns="http://test.ws.slacal.org/ws-slip" />
  </soap:Body>
</soap:Envelope>
```

**Response message:**

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <VerifyCredentialsResponse xmlns="http://test.ws.slacal.org/ws-slip">
      <VerifyCredentialsResult>
        <StatusCode>string</StatusCode>
        <StatusMessage>string</StatusMessage>
      </VerifyCredentialsResult>
    </VerifyCredentialsResponse>
  </soap:Body>
</soap:Envelope>
```

**Response parameters:**

Parameter	Data Type	Description
StatusCode	String	Indicates whether the credential has been successfully verified. The value "1" indicates success and "0" means failure.

Parameter	Data Type	Description
StatusMessage	String	A message describing the status of the credential verification if any error occurred during processing. "Method call successful." if the credential has been verified successfully.

#### 4.4.2. Upload Batch Filing Endpoint

The AMS must prepare a batch ZIP file that will contain all relevant information pertaining to the policies and transactions appropriate for batch submission. The ZIP file will then be submitted to the Upload Batch Filing endpoint. Upon completion of the batch filing, the API will provide the AMS with a value that uniquely identifies the batch submission attempt (submission number).

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

##### Request message:

```
POST /AMSBatchFiling.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://test.ws.slacal.org/ws-slip/UploadBatchFiling"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <AuthenticationHeader xmlns="http://test.ws.slacal.org/ws-slip">
      <SLABrokerNumber>string</SLABrokerNumber>
      <UserName>string</UserName>
      <APIKey>string</APIKey>
    </AuthenticationHeader>
  </soap:Header>
  <soap:Body>
    <UploadBatchFiling xmlns="http://test.ws.slacal.org/ws-slip">
      <FileName>string</FileName>
      <Data>base64Binary</Data>
      <Comments>string</Comments>
      <PreviewDataInSLIP>boolean</PreviewDataInSLIP>
    </UploadBatchFiling>
  </soap:Body>
</soap:Envelope>
```

**Request parameters:**

Parameter	Data Type	Description
FileName	String	The physical name of the file being submitted including file extension.
Data	Binary	The content of the policy submission as a base64Binary format
Comments	String	Comments for the batch filing
PreviewDataInSLIP	Boolean	Indicates whether user wants to check the policy data in SLIP before submission.

**Response message:**

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UploadBatchFilingResponse xmlns="http://test.ws.slacal.org/ws-slip">
      <UploadBatchFilingResult>
        <StatusCode>string</StatusCode>
        <StatusMessage>string</StatusMessage>
        <SubmissionNumber>string</SubmissionNumber>
      </UploadBatchFilingResult>
    </UploadBatchFilingResponse>
  </soap:Body>
</soap:Envelope>

```

**Response parameters:**

Parameter	Data Type	Description
StatusCode	String	Indicates if the request is successful or not. The value "1" indicates success and "0" means failure.
StatusMessage	String	A message describing the status of the request processing if any error occurred during processing. "Method call successful." if the request has been processed successfully.
SubmissionNumber	String	A value assigned for the batch submission.

#### 4.4.3. Check Status Endpoint

The check status endpoint will allow the AMS to obtain the status of a batch submission.

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

##### Request message:

```
POST /AMSBatchFiling.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://test.ws.slacal.org/ws-slip/CheckStatus"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <AuthenticationHeader xmlns="http://test.ws.slacal.org/ws-slip">
      <SLABrokerNumber>string</SLABrokerNumber>
      <UserName>string</UserName>
      <APIKey>string</APIKey>
    </AuthenticationHeader>
  </soap:Header>
  <soap:Body>
    <CheckStatus xmlns="http://test.ws.slacal.org/ws-slip">
      <SubmissionNumber>string</SubmissionNumber>
    </CheckStatus>
  </soap:Body>
</soap:Envelope>
```

##### Request parameters:

Parameter	Data Type	Description
SubmissionNumber	String	The submission number returned as the result of the batch submission.

##### Response message:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CheckStatusResponse xmlns="http://test.ws.slacal.org/ws-slip">
      <CheckStatusResult>
        <StatusCode>string</StatusCode>
        <StatusMessage>string</StatusMessage>
        <SubmissionNumber>string</SubmissionNumber>
      </CheckStatusResult>
    </CheckStatusResponse>
  </soap:Body>
</soap:Envelope>
```

```

    <Status>Uploaded or InvalidUpload or PendingUserAction or
PendingSLAReview or Closed or Returned</Status>
    <Errors>
      <SubmissionError>
        <ElementId>string</ElementId>
        <ElementType>Policy or Transaction</ElementType>
        <ErrorMessage>string</ErrorMessage>
      </SubmissionError>
      <SubmissionError>
        <ElementId>string</ElementId>
        <ElementType>Policy or Transaction</ElementType>
        <ErrorMessage>string</ErrorMessage>
      </SubmissionError>
    </Errors>
  </CheckStatusResult>
</CheckStatusResponse>
</soap:Body>
</soap:Envelope>

```

### Response parameters:

Parameter	Data Type	Description
StatusCode	String	Indicates if the request is success or not. The value "1" indicates success and "0" means failure.
StatusMessage	String	A message describing the status of the request processing if any error occurred during processing. "Method call successful." if the request has been processed successfully.
SubmissionNumber	String	The submission number returned as the result of the batch submission.
Status	Enumerated Value	<p>Identifies the status of the submission. The possible statuses are as follows:</p> <p><b>Uploaded</b> – The status indicates either (1) SLIP has identified and received an XML or AMS batch. The received batch will upload into SLIP automatically, or (2) An XML or AMS batch has passed schema validation and is in the Importer Queue awaiting import.</p> <p><b>Invalid Upload</b> – The status indicates either (1) An XML or AMS batch has failed schema validation and has been placed in the Responder Queue for notification, or (2)</p>



Parameter	Data Type	Description
		<p>SLIP has failed to import a batch that had been accepted for import. EAS has not accepted a batch.</p> <p><b>Pending User Action</b> – The batch was successfully uploaded to SLA and is pending SLA Review.</p> <p><b>Pending SLA Review</b> – The status indicates either (1) The batch has been placed on the EAS Exporter Queue and will upload from SLIP to SLA, or, (2) The batch was successfully uploaded to SLA and is pending SLA Review.</p> <p><b>Closed by SLA</b> – The Batch was invoiced by the SLA. (Note: Closed batches can contain Open Tags.)</p> <p><b>Returned by SLA</b> – The batch either was received and returned by the SLA or was not successfully uploaded. The SLA will notify users in either instance by both email and written letter.</p>
Errors	Array	<p>When a batch filing is rejected, an array of errors will be provided containing the issues that occurred during processing.</p> <p><b>SubmissionError:</b> Represents an error that occurred during processing. Each error identifies an element from the submitted data by custom element identifier and the element type (transaction or policy, for example) specified by the AMS. Batch filing errors are failures and cause the submission to be discarded.</p>

#### 4.4.4. Get Notifications Endpoint

During submission processing, notifications may be generated that can provide the user with feedback or recommendations for elements within the submitted data.

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

##### Request message:

```
POST /AMSBatchFiling.aspx HTTP/1.1
Host: serverhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://test.ws.slacal.org/ws-slip/GetNotifications"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <AuthenticationHeader xmlns="http://test.ws.slacal.org/ws-slip">
      <SLABrokerNumber>string</SLABrokerNumber>
      <UserName>string</UserName>
      <APIKey>string</APIKey>
    </AuthenticationHeader>
  </soap:Header>
  <soap:Body>
    <GetNotifications xmlns="http://test.ws.slacal.org/ws-slip">
      <SubmissionNumber>string</SubmissionNumber>
      <NotificationType>Alert</NotificationType>
    </GetNotifications>
  </soap:Body>
</soap:Envelope>
```

##### Request parameters:

Parameter	Data Type	Description
SubmissionNumber	String	The submission number returned as the result of the batch submission.
NotificationType	Enumerated Value	Allows the AMS to specify the type of notifications to be returned.

##### Response message:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
```

```

<GetNotificationsResponse xmlns="http://test.ws.slacal.org/ws-slip">
  <GetNotificationsResult>
    <StatusCode>string</StatusCode>
    <StatusMessage>string</StatusMessage>
    <SubmissionNumber>string</SubmissionNumber>
    <NotificationStatus>NotAvailable or AlertsExist or
NoAlertsExist</NotificationStatus>
    <Notifications>
      <Notification>
        <Type>Alert</Type>
        <ElementId>string</ElementId>
        <ElementType>Policy or Transaction</ElementType>
        <Message>string</Message>
      </Notification>
      <Notification>
        <Type>Alert</Type>
        <ElementId>string</ElementId>
        <ElementType>Policy or Transaction</ElementType>
        <Message>string</Message>
      </Notification>
    </Notifications>
  </GetNotificationsResult>
</GetNotificationsResponse>
</soap:Body>
</soap:Envelope>

```

### Response parameters:

Parameter	Data Type	Description
StatusCode	String	Indicates if the request is successful or not. The value "1" indicates success and "0" means failure.
StatusMessage	String	A message describing the status of the request processing if any error occurred during processing. "Method call successful." if the request has been processed successfully.
SubmissionNumber	String	The submission number returned as the result of the batch submission.
NotificationStatus	Enumerated Value	Indicates if notification is available depending on the batch status and if alerts exist when the notification is available. The possible notification statuses are as follows: <b>Not Available</b> – Alerts are not available now. The alerts will be available only when the batch status is Invalid Upload, Pending User Action, or Pending SLA Review. <b>Alerts Exist</b> – Alerts exist and the alerts will

Parameter	Data Type	Description
		be listed in the array of notification. <b>No Alerts Exist</b> – No alerts exist.
Notifications	Array	<p>When alerts exist, an array of notifications will be provided based on what occurred during processing.</p> <p><b>Notification:</b> Represents a notification entity (Alert) that occurred during processing. Each notification contains a message and fields necessary to identify the submitted data element with which the notification is associated.</p>

## 5. Batch Image File (BIF) Upload (Non-XML)

### 5.1. Description

The BIF Upload method allows brokerages to submit groups of document images. This method does not include alerts prior to user submission. Instead, users get a single email notification when a batch is submitted. BIF files are reviewed, processed, and filed as in any other batch method. Acceptable file types include PDF, PDF/A, PNG, TIFF, and JPG/JPEG.

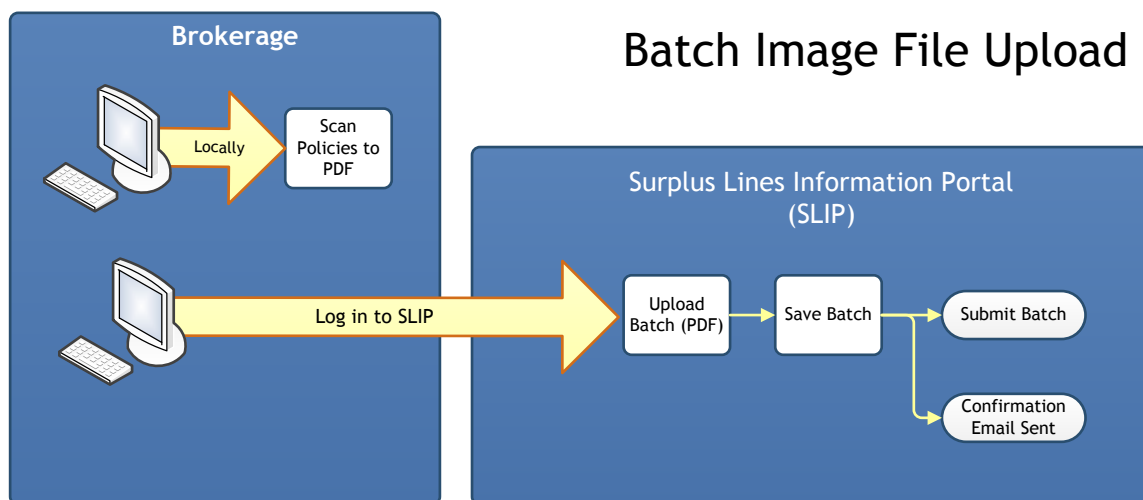
### 5.2. Pre-requisites

Brokerages may submit policy data using the BIF Upload method if the following requirements are met:

A SLIP user account must exist for a submitting brokerage. SLA California will create one master user account for each participating brokerage. The master user within each brokerage will have the ability to create other user accounts for that brokerage.

### 5.3. Process

This section identifies the steps required to create and submit policy information using the batch image file (BIF) submission process. The high-level representation of the process flow below indicates the systems involved.



### 5.3.1. Create New Batch

The first step in the BIF submission process is to gather the policies that will be in the batch and create images files. Scan the policies within your batch to a PDF format. You may scan policies into separate files, or consolidate your batch into a single PDF.

### 5.3.2. Log in to SLIP

Using a supported web browser, go to the SLA California SLIP website. Enter your username and password. This will establish a secure connection and validate your identity.

### 5.3.3. Upload and Submit BIF

From the SLIP home screen, click the Upload Batch Image File link. Enter the necessary details on the Create New Batch Wizard. From the Upload Documents to Batch panel, select the appropriate document type and click Browse. In the provided window, select the PDF batch file from your computer and click the Open button. Click the Upload button and save. Once the batch is saved, select Submit.



#### Create New Batch Wizard

SLIP >> Submission Type Selection (Step 1 of 3)

**Submission Type Selection**

Please select a submission type from the options below:

☐ Create Batch

Click "Create Batch" to create a batch with policies you have already created in SLIP.

☐ Bulk Upload (Advanced: XML Format Only)

Click "Bulk Upload" to upload policy data in bulk into SLIP. The Bulk Upload feature requires a ZIP archive containing the bulk policy data in a single XML file along with any additional related documents. You will have the ability to examine these policies and check for errors before submitting them to the SLA.

For details on the required XML format, [review the Bulk Upload procedures.](#)

☒ Upload Batch Image File (Broker Extranet Method)

Select this option to upload a group of document images into SLIP for submission to the SLA. This is similar to the Broker Extranet process. This method does not have the benefit of "early warning" alerts that are used for user review prior to submission, but are still reviewed, processed and filed by the Data Analysis Department at the SLA.

\*Please note that you will only get a single email notification when your batch is submitted to the SLA. There will not be any batch status updates for this option.

**Next >**

Figure 5 - Create New Batch: Upload Batch Image File



## Create New Batch Wizard

SLIP >> Batch Details (Step 2 of 2)

**Batch Details (Optional)**

Batch Source: Batch Image File

Broker Batch Date:

Broker Batch ID:

Late Filing Explanation:

Notes:

**All Associated Documents**

\*A batch coversheet is mandatory with your SLA filing. Please refer to the example on the SLA website: [Batch Coversheet](#)

Date Uploaded	Type	File Name	Remove
No documents are currently associated with this batch.			

**Upload Documents to Batch**

Files must be in one of the following formats: TIFF, PDF, PNG, PDF/A, or JPG/JPEG

Document Type

File Name

MULTIPLE

Browse...

Upload

< Back

Save

Figure 6 - Create New Batch Wizard



## View Batch

SLIP >> View Batch

< Back

**Batch Details**

SLIP Batch ID: 12021

Batch Source: BIF

SLA Batch ID:

Broker Batch ID: 123456

Late Filing Explanation:

Batch Status: NOT SUBMITTED TO SLA

Created Date: 4/28/2015 8:42:32 AM

Submission Date:

Broker Batch Date: 4/28/2015

Notes: Test

**All Associated Documents**

\*A batch coversheet is mandatory with your SLA filing. Please refer to the example on the SLA website: [Batch Coversheet](#)

Date Uploaded	Type	File Name
4/28/2015	MULTIPLE	<a href="#">batch-submission-manual-for-brokerages-v2-2-(pdf).pdf</a>

Delete Batch

Edit Batch

Submit

Figure 7 - View Batch & Submit

#### **5.3.4. SLA Validates the File**

After you successfully upload a batch in SLIP, the system will queue the submission for processing. When the system is ready to process the submission, the validation process will begin.

After submission, an e-mail will be sent to the submission contact identified with the batch. With the BIF method, the Data Analysis Department at the SLA will review the submission, but there will be no batch status updates issued.



## 6. Frequently Asked Questions

The following list identifies frequently asked questions from technical resources concerning the XML Batch Upload:

1. Do I need a SLIP account to submit a Batch file?

**Answer:** Yes, a SLIP account is required to submit policy data. If you are using the API method, you also must submit the SLA Broker Number and API key value in the SOAP header.

2. What is the user token? How can I generate a token?

**Answer:** The user token is a unique identifier that is created for a particular user. Users will log into SLIP to generate a token for their user account.

3. Can I use Excel to export a file to Batch?

**Answer:** The data contained within a batch submission must be in XML format. XML is a different way of storing data than Excel. XML is the leading standard for data exchange, providing several inherent benefits including data validation, structural enforcement, and platform independence. Please work with your technical staff to prepare your file appropriately.

4. What is the "SLABrokerNumber" element in the SOAP Header and XML Schema?

**Answer:** The Brokerage Number element refers to a unique identification number that is assigned to each brokerage by the Surplus Line Association of California.

5. What is the purpose of the "XML\_TransactionId" in the transaction element of the XML Batch Upload Method?

**Answer:** The Transaction ID is a unique alphanumeric value provided by the brokerage data management system and used to uniquely identify a policy transaction submitted using the manual batch file upload.

6. How can I generate a batch file from our data management system?

**Answer:** You will need to work with your IT staff to identify the best method to export data from your data management system in the required format.

7. Can I use the manual batch file upload method and also use the manual data entry method?

**Answer:** Yes, the system can handle this, but it is recommended you use only one method to avoid the possibility of duplicating filing submissions.

8. Can I edit a policy transaction that has been submitted through the manual batch file upload method?

**Answer:** Yes, you can edit transactions in SLIP that were submitted through the manual batch file upload method.

9. How often can I upload a batch?

**Answer:** There is no restriction on how often a batch may be uploaded. However, the Surplus Line Association of California does request that organizations with high volume please communicate with the Surplus Line Association of California to determine an appropriate rate of submission that will ensure a timely examination process.