Nevada Surplus Lines Association

Batch Submission Manual

SLIP

NEVADA SURPLUS LINES ASSOCIATION NSLA

Last Modified July 31, 2025

TABLE OF CONTENTS

1. Purpose

The purpose of this document is to provide technical information on the Nevada Surplus Lines Association (NSLA) batch submission process. The batch submission process can be used by brokerages to electronically submit policy data in bulk to NSLA. This document describes the batch submission process and includes technical specifications.

The intended audience for this document includes technical resources that will work with the brokerage to extract policy data from the brokerage's data management system for the purpose of creating submissions to NSLA. Technical resources typically include system architects, designers and developers with computer programming experience. It is assumed the reader of this document has an understanding of JSON (JavaScript Object Notation) and HTTP protocols.

Additional information on the batch submission process, including the JSON Schema and example upload files, can be found online within the SLIP section of NSLA's website (http://www.nsla.org).

2. Introduction

The Surplus Lines Information Portal (SLIP) is a web-based system that allows brokerages to electronically submit policy information to NSLA. Within SLIP, brokerages have two options to electronically submit policy data:

1. SLIP Manual Method - Data is entered manually through the SLIP website interface.
2. JSON Batch Submission Method - Data for one or more new policies (in the form of a JSON Document) is uploaded to SLIP.

The help files and video tutorials contained within SLIP provide instructions for submitting policy data using the Manual Method and will not be addressed in this document. Currently only new policies can be filed using batch files. To edit existing policies, you may only manually edit using the SLIP web interface.

2.1 The JSON Batch Submission Method

The JSON Batch Submission Method allows brokerages to submit policy and transaction data for multiple policies at once in a single batch process. This process will especially benefit brokerages that file a large amount of policy data with NSLA since a single JSON file may contain information for multiple policies.

Brokerages that store data in a centralized data management system can make use of the JSON Batch Submission Method. The following list provides a high-level list of the steps contained within the batch submission process:

1. The brokerage will generate a JSON file containing the policy data they wish to submit to NSLA. Typically, the JSON file will include policy data that was added or modified within a specified date range or since the last JSON batch submission.
2. The brokerage may also include any associated policy documentation (PDF is the preferred format) that is associated with the policy data contained within the JSON file. The policy documents will be referenced by filename within the JSON file.
3. The brokerage will create a zip file containing the JSON file and the associated policy documentation (again, referenced by filename within the JSON file).
4. The brokerage will log in to SLIP to upload the zip file containing the JSON file and all associated documents.

2.2 Prerequisites for the JSON Batch Submission Method

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

1. A SLIP user account must exist for a submitting brokerage. NSLA will create one administrative user account for each participating brokerage. The administrative user within each brokerage will have the ability to create other user accounts for that brokerage.
2. SLIP requires a modern web browser, such as Google Chrome or Microsoft Edge.

3. The JSON Batch Submission Process

This section identifies the steps required to create and submit policy information using the JSON batch submission process. The following section (Section 4 - Batch File Guidelines) outlines the requirements of the JSON file.

3.1 Create Batch File

The batch file is a zip file that contains one JSON file (including policy data on one or more policies) and policy documentation (electronic copies of Dec Sheets, Supplemental Dec's, Forms List, etc.). The creation of the batch file will require the involvement of a technical resource that is familiar with JSON and the data management system in use by your brokerage. There are several different data management systems in use 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 in 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 NSLA is required.

Once the criteria to extract policy data is identified for your data management system, a technical resource must create the JSON file that contains the policy data. For details on the required format and structure of the JSON file, please refer to Section 4 – Batch File Guidelines. Also, a JSON schema is available on the NSLA website under the SLIP – Batch Reporting Procedures section. The JSON schema identifies technical constraints on the content and structure of the JSON file. Also, the JSON schema can be used to validate the JSON file prior to submission.

In addition to submitting policy data, the brokerage may also include policy documentation associated with the policy data contained within the JSON file. The policy documentation contained within the batch submission must be individually named for each policy transaction (New Business, Additional Premium, Return Premium, etc.) and referenced by filename within the appropriate section of the JSON file for each transaction. The file formats that can be accepted within the batch file include PDF, DOCX, XLSX, TIF, TIFF, JPG, JPEG. PDF is the preferred format.

Once the JSON file is created and the policy documents have been identified and/or extracted from the data management system, all the files should be included in one zip file for submission to NSLA.

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

3.2 Log in to SLIP

In your browser, go to the following website: https://slip.nsla.org. Enter your username and password. This will establish a secure connection and validate your identity.

If you are a surplus lines brokerage and you do not have a username and password, please contact NSLA at (775) 826-7898. NSLA 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 Upload and Submit the Batch 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 JSON file and associated policy documents (PDF, DOCX, XLSX, TIFF, or JPG). Submit the ZIP file for upload.

3.4 SLIP Validates the File

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

3.5 Monitor the Batch Submission Status

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

1. Received – The file has been successfully uploaded into SLIP and is waiting for processing.
2. Processing - The JSON file is currently being processed by the batch filer process.
3. Submission Rejected – The JSON file failed the validation process and cannot be imported. An email will be sent identifying the submission error(s).
4. Submission Accepted – The JSON file was successfully processed. An email will be sent confirming a successful submission.
5. Submission Errors – There was an unexpected error processing the batch submission. The support team has been notified, please give time for us to investigate and determine the issue before resubmitting. Someone will be in contact with you to explain what corrections are necessary.

3.6 Batch File is Accepted or Rejected

If the file is received, no further action is required. As mentioned in section 3.5, you may monitor the JSON import process on the Batch Submission page.

If the file has been rejected, 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 NSLA at (775) 826-7898.

4. Batch File Guidelines

4.1 Batch File Size and Structure

The batch zip file size cannot exceed 200 MB. The root of zip file should contain one JSON file and the policy documents associated with the filings in the JSON file. The zip file should not contain any subfolders.

4.2 Batch File Name

The file name is limited to 250 characters. There is no required naming convention, however, it recommended that you create filenames that make it easy to maintain and track your submissions. We suggest that you include the submission date and time in the file name. For example, 20240605_0930_Batch.zip (*date_time*_Batch.zip or *CCYYMMDD_HHMM*_Batch.zip) would indicate the batch was created on 06/05/2024 at 9:30 AM.

4.3 JSON Encoding

The complete JSON file must conform to standard JSON formatting and conform to the batch filing JSON schema before it is submitted. Failure to do so will result in processing delays and rejections. A typical rejection error is an "Invalid JSON" message.

Please review the JSON schema in the SLIP Batch Filing section of the www.nsla.org website.

4.4 Initial File Validation and Rejection

If the document fails ANY of the validations identified below, the ENTIRE batch file will NOT be accepted.

The following items describe the initial file validation process completed by SLIP.

1. Parse the document and check that the document is well-formed.
2. Check the JSON file document against the JSON Schema definition file.
3. Check the length of all data elements to ensure they do not exceed maximum lengths.
4. Check that all required data elements are present (e.g. policy number, transaction type, coverage code, insurers, etc.).
5. Check that values of the specified elements comply with the detailed JSON document requirements (Section 4.7) and the JSON schema (Section 4.8).
6. Check that the document file names (policy documentation) contained within the zip file match those listed in the JSON file.
7. Check that all associated policy document types are PDF, DOCX, XLSX, TIFF, or JPG. Any other format will not be accepted.
8. Check that each associated policy document (contained in the zip file) is less than or equal to 3 MB.

If any of the file 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 data file and resubmit it through SLIP.

4.5 Policy Data Validation and Rejection

If the policy data contained within the JSON file fails ANY of the following validations, the ENTIRE batch file will NOT be accepted.

The following items describe the secondary validation process completed by SLIP.

1. Check for complete policy data. Data must be included for all required policy fields.
2. Check for a valid brokerage and brokerage identification number. The brokerage identification number is the surplus lines brokerage license number. The brokerage license number can be found online at https://www.nsla.org under the Resources, Brokerage List section.
3. Check to ensure the JSON batch isn't attempting to modify a previously submitted policy. Currently, only new policy data is accepted by the batch filing process. To edit existing SLIP policy data, you must do so manually through the SLIP website.
4. Accept and/or reject file. An e-mail will be sent to the submission contact to confirm the acceptance or rejection of the file. If the file has been rejected, the message will contain a detailed description of the problem(s) and instructions to correct and resubmit the file. The user must correct the data file and resubmit it through SLIP.

4.6 JSON Schema Specifications

The JSON batch submission process will allow a brokerage to create new policy and transaction data through the batch submission automated process. Documentation on JSON Schema can be found at https://json-schema.org/ . Review the schema.json schema found at https://www.nsla.org which will have information about which version of JSON Schema is used, and how the batch filing object should be structured.

4.7 Table of JSON Fields

The table below outlines the specific values for the required elements in a JSON schema that is prepared for the Batch Submission Method. In the table, every element is individually addressed, and a sample of the JSON Structure is provided. The JSON Structure provides an example of the hierarchy and structural format that the submitted JSON will be validated against. The JSON structure is followed by a brief description of the element and the element's occurrence and length requirements

For additional detail on the required format for the JSON and/or the allowable values contained within the JSON, please refer to the JSON Schema file and batch upload examples contained within the SLIP – Batch Reporting Procedures section of the NSLA website (http://www.nsla.org).

JSON Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
{	Opening JSON object symbol, for the entire Filing object.	1	1	-	-
"SubmissionContact": {	Contact information of the person who will submit the JSON file	1	1	-	-
"FullName": "John Jason Smith Jr.",	Full name of the submission contact	1	1	3	120
"Email": "smithj@emailaddress.com",	Email address for the submission contact	1	1	1	250
"Address": {	Address for the submission contact	1	1	-	-
"Street": "**123 Main Street**",	Street Address (line 1) for the submission contact	1	1	1	100
"StreetLine2": "**Suite 1001**",	Street Address (line 2) for the submission contact	0	1	0	100
"City": "**Reno**",	City for the submission contact	1	1	1	50
"State": "**NV**",	State for submission contact (official state abbreviations)	1	1	2	50
"ZipCode": "**89509**"	Zip code for the submission contact (5 digits + optional 4)	1	1	5	10
}	Ending Element for Address				
}	Ending Element for Submission Contact				
"Brokerage": {	Reporting brokerage details	1	1	-	-
"LicenseNumber": "**1111**",	Brokerage identification number (License #). See the Brokerage List on the NSLA website for details (www.nsla.org)	1	1	1	7

JSON Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
"BrokerageName": "**ABC, Inc.**"	Name of the reporting brokerage	1	1	1	100
}	Ending Element for Brokerage				
"Policies": {	Details about policies placed within the reporting period	1	unbound	-	-
"PolicyNumber": "**PLC234-2406-567**",	Policy number	1	1	1	50
"Broker": {	Details about Broker assigned to the policy	1	1	-	-
"FirstName": "**Mike**",	First name of the broker	0	1	0	50
"LastName": "**Johnson**",	Last name of the broker	0	1	0	50
"LicenseNumber": "**123456**"	WAOIC number of the broker	1	1	1	7
}	Ending Broker Element				
"DescriptionOfCoverage": "**Test**",	Description of coverage regarding the policy	1	1	1	250
"Insured": {	Individual or business that is insured	1	1	-	-
"NameInsured": "**John Appleseed**",	Name of the insured	1	1	1	295
"InsuredAddress": {	Address of the insured	1	1	-	-
"Street": "**40 N Purple Sage Ave**",	The first line of the street address of the insured	1	1	1	100
"StreetLine2": "**Unit #76021**",	Second line of the street address of the insured	0	1	0	100
"City": "**Alamo**",	City of the insured	1	1	1	50
"State": "**NV**",	State for the insured (Official state abbreviations)	1	1	2	50
"ZipCode": "**89001**"	Zip code of the insured (5 digits + optional 4)	1	1	5	10
}	Ending Address Element				
"IsInsuredAddressSameAsLocation": **0**,	When set to 1, do not include LocationOfRiskAddress *	0	1	-	-
"LocationOfRiskAddress": {	Address of the insured risk.	0*	1	-	-
"Street": "**9631 Old Mill St**",	The first line of the street address of the risk	1	1	1	100
"StreetLine2": "**Bldg. 1**",	Second line of the street address of the risk	0	1	0	100
"City": "**Rachael**",	City of the insured	1	1	1	50
"State": "**NV**",	State for the insured (Official state abbreviations)	1	1	2	50

JSON Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
"ZipCode": "**89001**"	Zip code of the insured (5 digits + optional 4)	1	1	5	10
}	Ending Address Element				
}	Ending Insured Element				
"Insurers": [	List of insurers covering the policy	1	1	-	-
{	An insurer covering the policy.	1	unbound	1	1
"InsurerNAIC": "**AA1234567**",	Insurer NAIC number	1	1	1	9
"InsurerName": "**INSURER, Inc.**",	Name of the insurer	1	1	1	100
"AllocationPercentage": **100**	Percentage of coverage by insurer	1	1	-	-
}	Ending Insurer Element				
],	Ending Insurer List Element				
"IsMasterPolicy": **0**,	Whether the policy is a master policy	1	1	-	-
"isMultiState": **0**,	Whether the policy is a multi-state policy	1	1	-	-
"IsCourtesy": **0**,	Whether the policy is a courtesy filing	1	1	-	-
"isLayered": **0**,	Whether the policy is a layered risk	1	1	-	-
"IsParticipating": **0**,	Whether the policy is a participating policy	1	1	-	-
"isEcp": **0**,	Whether the policy is for an exempt commercial purchaser	1	1	-	-
"Declinations": [	Declinations are only required for certain coverage codes *	0*	1	-	-
{	Declination object for an ineligible for export coverage code*	3	3	-	-
"Admitted": "**Denizens Property Insurance**",	Admitted Insurer	1	1	1	100
"Underwriter": "**Jim Halphart**",	Underwriter	1	1	1	100
"ReasonCode": **4**	Reason Code (see schema for code lookup information)	1	1	-	-
}	Ending Declination Element				
],	Ending Declination List Element				
"CoverageCode": "**0300**",	Policy coverage code	1	1	1	4

JSON Structure	Description	Occurrence		Length	
		Min	Max	Min	Max
"Notes": "**Rare merchandise risk**",	Notes about policy	0	1	1	4000
"Transactions": [	Transactions list associated with a policy	1	1	-	-
{	Transaction object	1	unbound	-	-
"TransactionType": **1**,	Transaction type (See schema for code lookup information)	1	1	-	-
"Premium": **1000.00**,	The amount charged/returned to the insured minus any fees. May be positive or negative	1	1	-	-
"Fees": **10.00**,	Fees charged/returned to the insured. May be positive or negative	1	1	-	-
"EffectiveDate": "**20240701**",	Effective date of the transaction. (CCYYMMDD)	1	1	8	8
"ExpirationDate": "**20250701**",	Expiration date is required for an *initiating transaction	0*	1	8	8
"Documents": [	Attached document list in batch submission	0	1	-	-
{	Attached document information object	1	unbound	-	-
"DocumentFileName": "**insuredBinder23629.pdf**"	Identifies the filename of the associated document contained within the submission. Include the full filename and extension.	1	1	1	250
}	Ending Document Element				
],	Ending Document List Element				
}	Ending Transaction Element				
]	Ending Transaction List Element				
}	Ending Policy Element				
]	Ending Policy List Element				
}	Ending Filing Element				

4.8 JSON Schema

An additional JSON Schema sample is available on the NSLA website (http://www.nsla.org) under the SLIP- Batch Reporting Procedures section.

4.9 Frequently Asked Questions

The following list identifies frequently asked questions from technical resources concerning the JSON 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 using the JSON Batch Upload Method. We recommend obtaining a test account when first getting started with creating batches.*

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

   Answer: *The data contained within a batch submission must be in JSON format. JSON is a different way of storing data than Excel. JSON 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.*

3. What is the "Declinations" element in the JSON Schema? Where can I find this information?

   Answer: *Declinations are required for coverage codes that are ineligible for export. These all start with the character "0". When filing using an ineligible for export coverage code, you must include 3 declinations. Contact your local stamping office for more information.*

4. When is ExpirationDate required?

   Answer: *A transaction expiration date is only required on initiating transactions, such as New Business, Renewal, and Reinstatement. Transaction dates affect the policy effective and expiration dates. This field is ignored for all non-initiating transactions.*

5. Can the Batch Upload Method modify previously submitted transactions?

   Answer*: At this time, batch uploads may only create new policy data. Existing policy data must be modified through the SLIP web user interface.*

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. There are several resources available on our website providing details about the Batch Upload Method including a Batch Reporting Procedures Manual, example batch files, and an JSON Schema.*

7. Can I use the Batch Upload Method and also use the Manual Submission 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 manually update a policy transaction that has been submitted through the JSON Batch Upload Method?

   Answer: *Yes, you can manually update transactions in SLIP that were submitted through the Batch 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 Nevada Surplus Lines Association does request that organizations with high volume please communicate with the Nevada Surplus Lines Association to determine an appropriate rate of submission that will ensure a timely examination process.*

10. Can I specify county and country in address information?

    Answer: *These are optional fields in the schema that you may use, although there are some limitations to them. Currently, USA and Canada are the only two countries you may choose. County is retrieved by zip code, so any value you give may get replaced by the county assigned to that zip code. Also, state must be a USA 2 letter state abbreviation, but when filing for Canada, state can be the full province name.*

11. How do I use the "IsInsuredAddressSameAsLocation" indicator?

    Answer: *When this attribute is set to 0, you must include both the InsuredAddress and LocationOfRiskAddress JSON objects. However when the attribute is set to 1, you must omit the LocationOfRiskAddress. The bulk filer will automatically use the InsuredAddress information for both address'.*

12. How do I interpret rejection responses?

    Answer: *Rejection responses are custom tailored to your batch submission. Each error is composed of three parts, an indication of where the error is located, what the error is or what the bad value is, and an error message. Location is sometimes indicated by numbers, such as policy 0, policy 1, transaction 0, transaction 1. Know that these numbers start at 0, so policy 1 would be the second policy in the JSON batch.*

13. What characters are allowed in batch submission data?

    Answer: *The system will automatically strip any prohibited characters without error. However, if all characters are prohibited, you may get a blank field error. Generally, any numbers, letters and symbols are allowed. Accent marks will be translated to the closest equivalent English alphabet letter.*

14. Where can I find a list of coverage codes?

    Answer: *Please visit www.nsla.org and in the Resources section of the left menu, visit either the Export List or Ineligible for Export links. Note that Ineligible for Export coverage codes require a declinations section.*

4.10 Additional JSON Information

JSON creation software may help you examine and work within the parameters of the JSON schema. You can find validator tools on the JSON Schema official website at https://json-schema.org/implementations .

The following websites contain valuable information regarding JSON schema, as well as some information concerning JSON tools.

- https://json-schema.org/
- https://docs.json-everything.net/