ELTO Employers' Liability Tracing Office

ELD NextGen Policy Management

Summary of changes

11th September 2026
Version 1.2

Contents

1 Purpose of this Document 2

2 At a Glance 2

2.1 The headline changes 2

2.2 Why we are doing this 2

3 How data is submitted 3

3.1 Two methods 3

3.2 File formats 3

3.3 Removed legacy artefacts 4

4 The data model — what changes 4

4.1 Single policy record 4

4.2 The compend key has changed 6

4.3 Renamed fields 6

4.4 Removed fields 6

4.5 Address structure 6

4.6 New capabilities 7

5 Operational changes 7

5.1 Categorisation 7

5.2 Ownership and permissions 8

5.3 Organisation types 8

5.4 Time to Supply 8

5.5 Upload ordering 8

6 Migration and portal 9

6.1 Data migration 9

6.2 ELD access 9

7 Technical Implementation References 9

1 Purpose of this document

This document explains the changes coming to the ELD policy data submission process. It is intended for product owners, integration leads, business analysts, and engineering managers who need to understand what is changing, why, and what their teams will need to plan for.

Technical implementation is supported by the ELD Getting Started Guide, ELD Policy Schema Guide and ELD File Transfer Guide. The Developer Portal contains the authoritative endpoint specification; the earlier API Outline Specification should be treated as superseded planning material once the refreshed full API specification is issued.

2 At a glance

The ELD system is moving from a file-drop SFTP transfer to an HTTPS API. Policy submissions will remain for individual policy updates via the ELD portal. Alongside the channel change, the data model has been simplified, validation has been made simpler and clearer, and a portal will provide a UI for direct policy management.

2.1 The headline changes

Area	Today	After change
Submission channel	SFTP file drop with proprietary whitespace encoding	HTTPS API (direct JSON or asynchronous file upload orchestrated by the API)
File formats	Custom proprietary format	Standard JSON (RFC 8259) or pipe-delimited text (Excel-compatible)
Validation feedback	Asynchronous, file-level	Synchronous for direct submissions; structured per-error reporting for files via the ELD or API.
Policy hierarchy	Parent / child employer relationship	Single policy record with all employers at equal standing
Compend / Policy key	Insurer + Policy No. + BrokerDAReference + Cover dates + Employer name + Policy type	Insurer ID + + policy holder name + Policy No. + Supp. Ref + Cover start Date
Categorisation	Separate uploads for historic vs current	Combined endpoints — categorisation derived from data
Ordering	File sequence numbers	Upload timestamp (with upload ID tiebreaker)
Portal access	Limited	Direct policy submission, status, validation results

2.2 Why we are doing this

The ELD NextGen programme aims to deliver a more robust, efficient, and future-ready platform for managing Employers' Liability data. With a particular focus on improving how you submit, maintain, and interact with policy information.
Our key objectives are to:

- Improve reliability and data quality – reducing errors, duplication, and ambiguity in policy data through better validation and clearer structures.

- Simplify and streamline policy submission through more efficient and scalable upload mechanisms that reduce manual effort and operational friction.
- Enhance auditability and traceability through clearer visibility of submissions, changes, and outcomes to support compliance and internal processes.
- Reduce long-term operational burden by adopting more robust and sustainable approaches that minimise ongoing support issues
- Support future-facing approaches (e.g. APIs) that can better integrate with your systems over time.
- Provide a managed and realistic transition – working collaboratively with you to ensure any changes are practical to adopt, with sufficient time for testing and onboarding.

3 How data is submitted

3.1 Two methods

Policy data will be submitted via HTTPS API. There are two methods, each suited to a different use case.

Method	When to use	Validation behaviour
Direct **JSON submitted in the request body.**	Smaller submissions where you want immediate confirmation. Suitable for online integrations and incremental updates.	Synchronous and all-or-nothing. If any policy fails validation, the entire submission is rejected and no policies are ingested.
File upload **Two-step upload via secure SAS URL.**	Larger batches and existing batch processes. Suitable for migrations, bulk catch-up, and scheduled overnight loads.	Asynchronous and per-policy. Valid policies are ingested; invalid ones are rejected individually. Errors are retrieved via the status endpoint.

NOTE: There is no minimum pass rate for file submissions — a submission is not failed because a certain proportion of records are invalid. The system processes what it can and reports what it cannot.

3.2 File formats

Two file formats are supported for file-based submissions:

Format	Extension	Notes
JSON	.json	Array of policy objects, mirroring the API request body. Recommended for most new integrations. Limited to 2MB per JSON submission.
Pipe-delimited text	.txt	Flat, row-per-ERN structure. Exportable from Excel with a small device-setting adjustment. Suitable for organisations whose source data is held in spreadsheets or seeking large bulk uploads.

Why this matters
The old SFTP format used proprietary whitespace encoding. The new formats are entirely standard - JSON follows the RFC 8259 specification and the pipe-delimited text format follows Excel conventions for quoting and escaping. No special encoding tooling is required.

3.3 Removed legacy artefacts

The following elements from the old SFTP format are no longer required:

- **File sequence numbers** — upload ordering is now derived from the upload timestamp.
- **Site IDs** — are no longer supplied. The insurer is selected by insurerId in the API path and is checked against the authenticated credential.
- **Header and trailer records** — control metadata is in the API request, not the file body

4 The data model — what changes

4.1 Single policy record

The most significant structural change is the **removal of the parent/child policy hierarchy**. Previously, a policy was modelled as a primary (parent) employer with one or more child employers attached. Going forward, a policy is a single record containing all covered employers as equal entries.

Example Schema
The following is an example of the schema for a policy submission. Full details of the schema and mapping to the legacy schema can be found in the Policy Schema Guide.


```
[
  {
    "policyNumber": "EMP-2024-001",
    "supplementaryPolicyReference": "BROKER-REF-12345",
    "coverStartDate": "2024-01-01",
    "coverEndDate": "2024-12-31",
    "policyholderName": "Acme Corporation Ltd",
    "firstClaimSettlementDate": "2024-02-01",
    "originalInsurerName": "Example Insurer Ltd",
    "firstSuppliedAt": "2024-01-15",
    "employers": [
      {
        "name": "Acme Corporation Ltd",
        "erns": [
          "123/AB45678"
        ],
        "crn": null,
        "adjustedCoverStartDate": null,
        "adjustedCoverEndDate": null,
        "hmrcErn": "123/AB45678",
        "legacyPolicyId": "1001",
        "multilineAddress": {
          "addressLine1": "123 Business Park",
          "addressLine2": "Suite 100",
          "cityTown": "London",
          "county": "Greater London",
          "country": "United Kingdom",
          "postCode": "SW1A 1AA"
        }
      },
```

```
      {
        "name": "Acme Manufacturing Division",
        "erns": [
          "456/CD90123"
        ],
        "crn": null,
        "adjustedCoverStartDate": null,
        "adjustedCoverEndDate": null,
        "hmrcErn": "456/CD90123",
        "legacyPolicyId": "1002",
        "multilineAddress": {
          "addressLine1": null,
          "addressLine2": null,
          "cityTown": null,
          "county": null,
          "country": null,
          "postCode": null
        }
      },
      {
        "name": "Acme Retail Division",
        "erns": [
          "789/EF45678"
        ],
        "crn": null,
        "adjustedCoverStartDate": "2024-03-01",
        "adjustedCoverEndDate": null,
        "hmrcErn": "789/EF45678",
        "legacyPolicyId": "1003",
        "singleLineAddress": "789 Retail Park, Birmingham, B1 1AA"
      }
    ]
  }
]
```

Practical implications:

- Each policy is a single record containing all covered employers under a named policy holder.
- All employers are held under the policy and all have equal standing — there is no concept of primary or parent.
- Each employer's address must be supplied. Where not known this can be the Policy holder address. Each employers ERN(s) and CRN can be supplied individually and remain optional.
- Whole-policy uploads are required every time. Each submission must include the complete policy including every employer. If a policy with a matching key already exists, it is fully overwritten by the new submission. The new submission is treated as the complete replacement for the policy with the same key, so missing employers are treated as intentionally removed from that policy.
- In the new model, validation applies to the policy record. For direct API submissions, if one employer in a policy fails validation, the whole direct submission is rejected. For file uploads, the affected policy is rejected, not a separate former parent/child transaction.

Implementation note

JSON submissions reflect this structure explicitly with a nested `employers` array. Pipe-delimited text submissions allow for row-per-ERN, but each row group keyed on the policy fields represents a single policy record.

4.2 The compend key has changed

The policy key, formerly compend / composite key has changed. Three former key components have been removed: cover end date, employer name and policy type. Together with the parent/child removal, this means a single policy can now legitimately cover several employers across multiple sites without artificial duplication.

Note: The full policy key will be confirmed in the full API guidance.

Field	Old key	New key
Insurer ID	✓	✓
Policy Holder Name	✓	✓
Policy Number	✓	✓
Supplementary Policy Reference (formerly BrokerDARef)	✓	✓
Cover Start Date	✓	✓
Cover End Date	✓ (partial)	**Removed**
Employer Name	✓	**Removed**
Policy Type	✓	**Removed**

4.3 Renamed fields

Old name	New name	Why renamed
EffectiveDate	firstClaimSettlementDate	Aligns with ICOBS terminology. Also moves from employer level to policy level. This field is used when a claim is made on a policy.
BrokerDARef	supplementaryPolicyReference	The purpose of this field has not changed and should be used to hold any supplementary reference often provided by a Broker or Delegated Authority.

4.4 Removed fields

The following fields are no longer required:

Field	Reason
DummyPolicy	No longer required.
PolicyType	Removed as part of consolidating employers into a single policy.
ERNExempt	No longer required.
SiteId	FSNs have been removed from the model.
AddressTypeLookUpId	Address type is determined implicitly from which fields are populated.
addressLine3 / addressLine4	Address structure simplified — see below.

4.5 Address structure

The supported address structure is: `addressLine1`, `addressLine2`, `cityTown`, `county`, `country`, `postCode` — or alternatively a single `singleLineAddress`. During migration, address lines 2, 3, and 4 from the old system will be combined into a single `addressLine2`. Suppliers may re-upload data in any preferred structure after migration; this does not affect system function or auditing.

4.6 New capabilities

4.6.1 Original Insurer

The Original Insurer is no longer submitted as an identifier and is is replaced with a free text field to provide the name of the original insurer. This should be the name of the insurer underwriting the policy at the time of issue as defined by the ICOBs requirement. Where the policy is historic and the Original Insurer is unknown. then "Unknown" may be used.

MIGRATION NOTE: During migration the current Original Insurer ID will be transformed into the name of the Insurer or set to Unknown if not known.

4.6.2 Multiple ERNs per employer

The erns field is now an array, allowing multiple Employer Reference Numbers per employer. In the old system, each employer could have at most one ERN unless they were added in the portal.

MIGRATION NOTE: Migration will carry over the master ERN per employer where one exists. **Additional ERNs will not be automatically migrated** — suppliers wishing to add further ERNs to migrated employers will need to re-supply.

4.6.3 Company Registration Number (CRN)

A new optional `crn` field has been added at employer level. This corresponds to the Company Registration Number. If you hold this value, please supply it.

Note: CRN is optional. If you do not hold CRN in your source systems, you may leave it blank. Supplying a blank CRN will not cause validation failure.

4.6.4 Insurer Ref

A new optional free text field called Insurer Ref has been added to the Policy Schema. This can be used to hold any internal reference that would assist you. The Insurer Ref is not used by ELTO or exposed in the policy search.

5 Operational changes

5.1 Categorisation

Separate files are no longer required for different categories of policies (e.g. historic, current). The new system uses a combined endpoint for all policies. Categorisation is determined automatically from the data — primarily the policy cover dates and the firstClaimSettlementDate where applicable. All policies can be submitted together in a single file or request.

5.2 Ownership and permissions

Policies are **owned by the last supplier who uploaded them**:

- **Insurers** can view all of their own policies regardless of who uploaded them.
- **Delegated authorities (DA)** can only view policies they themselves uploaded. If another supplier subsequently uploads a new version of a policy, the DA loses visibility of it.

When submitting, the supplier is inferred from the credentials used; the insurer is identified explicitly in the path of the request (which may be the insurer directly or a DA acting on their behalf), and the `insurerId` identifies the insurer who owns the policy. Where the insurer is supplying on their own behalf, these will be identical.

Within the ELD portal you will be able to view, edit and delete individual policies your organisation owns. Via the API you can edit and delete policies your organisation owns or that another supplier (i.e. DA) has uploaded, if the changes being made are to policies with existing keys.

5.3 Organisation types

The legacy ELD requires a supplier-type field on each submission. In the new system, every organisation has a type stored against its record. Where an insurer is also a DA, these will be handled in the system as separate organisations.

As a result, the `supplierId` and `insurerId` convey type information implicitly — no separate type field is needed.

5.4 Time to Supply

The Time to Supply (TTS) calculation has been updated to align with the new data model:

- TTS is based on the **policy-level key**: policy number, supplementary policy reference, and cover start date.
- **Changes within a policy** (e.g. updated employer details, addresses, policyholder name) do **not** trigger a new TTS measurement.
- **New key combinations** are treated as new policies with TTS calculated from first supply.

5.5 Upload ordering

Upload order is determined by **upload timestamp**, with upload ID used as a tiebreaker where timestamps are equal. This defines the order in which policy updates are applied and replaces the old file sequence number mechanism.

Note: Upload timestamp replaces file sequence numbers. Submitting an older full-policy version after a newer one may overwrite the newer policy state if it has the same key and a later upload timestamp.

6 Migration and portal

6.1 Data migration

All existing policy data will be **automatically migrated** from the current system to the new system. Insurers do not need to re-submit their existing data.

A small number of data adjustments will be made as part of migration to align with the new data model. These include:

- Fields previously stored per employer will be consolidated to policy level where values are consistent across the policy.
- Address lines 2, 3, and 4 from the old system will be combined into a single addressLine2.
- Master ERN per employer will be carried over; additional ERNs will not.

What you'll receive
More precise detail on migration changes — and the logistics of how migration will run — will follow before they take place. This will include guidance on reconciliation and data cleansing.

Suppliers will have an opportunity to review and re-upload after migration if a different address structure or richer ERN list is preferred.

An Assurance Document confirming the migration outcome will be provided for audit purposes.

6.2 ELD access

Policy management functionality will also be available via the ELD portal (web interface):

Available via portal:

- Single policy submission (create, update and delete)
- Processing status for portal submissions and API submissions, including validation results.

Not available via portal:

- Ability to upload a file of policies: This is only available via API

7 Technical Implementation References

Please use the following as references in developing your integrations and preparing for the change:

- Use the **Developer Portal** as the authoritative endpoint specification.
- Use the **ELD Policy Schema Guide** for field purpose and legacy-to-NextGen mapping.

- Use the **ELD File Transfer Guide** for pipe-delimited layout and file preparation.
- Use the **ELD Getting Started Guide** for portal access, credentials, test approach and published limits.
- Use the **Developer Portal bearer-token guidance** for certificate-based token generation.