

Phoenix Stocks

V1.0

1. Overview

Phoenix Stocks is an Enfocus Switch flow element that lets you read from and write to your **Esko Phoenix** stock library directly from a Switch flow. Anything you can do by hand in Phoenix's library — adding a new stock, attaching a grade, recording sheet sizes, deleting old materials — you can do automatically as part of a job's journey through Switch.

A single Phoenix Stocks element can do any one of **21 actions**: it can **Find, Check for, Create, Edit, or Delete** a Stock, Grade, Sheet, or Roll. It can also **Duplicate** a Stock. You pick the action from a dropdown in the element's property panel; the rest of the form changes to show only the fields that action needs.

Every job that leaves the element comes out one of three connections — **Success, Warning, or Error** — and (for almost every action) carries a JSON file attached that records exactly what the element did. You can wire your flow off any of those three outputs and read the JSON file in a downstream script if you need richer logic.

***Note:** This manual covers what the element does and how to use it. It does **not** explain Phoenix itself — for questions about what a "grade" means in your imposition workflow, what cost units to choose, or how stock types affect press selection, please refer to Esko's Phoenix documentation.*

2. Quick start

Drop a **Phoenix Stocks** element onto your flow. It needs at least one incoming connection and at least one outgoing connection.

Set the two connection-server properties at the top of the property panel:

Field	What it is	Default
URL	Hostname or IP address of your Phoenix server.	localhost
Port	TCP port Phoenix listens on.	8022

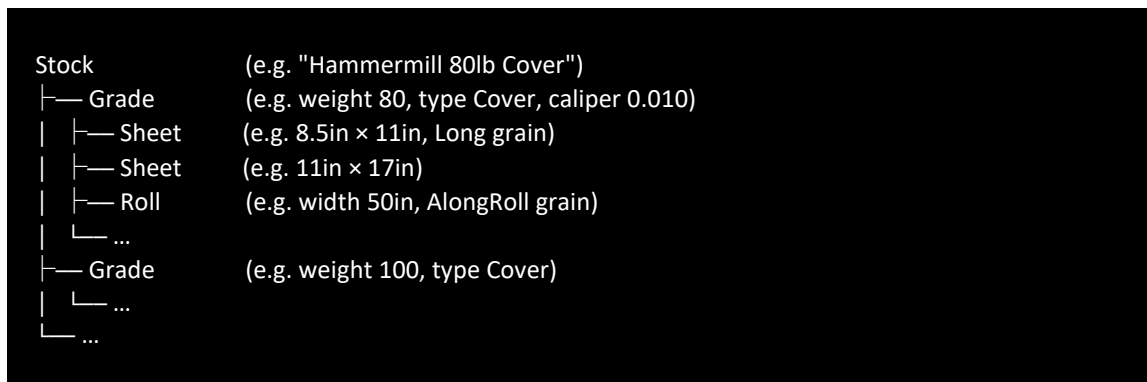
Then choose an **Action** from the dropdown. The form below will update to show only the fields that action needs. Fill them in, save the flow, and run a job through it.

When the job comes out the Success (or Warning) port, look in the job's **Datasets** — you'll find a file called `<jobName>-result.json` with the structured result. When it comes out the Error path, you'll usually find `<jobName>-error.json` with the underlying Phoenix response (when Phoenix returned an actual API error).

3. Concepts you'll use in every action

3.1 Stocks, Grades, Sheets, Rolls

Phoenix organizes substrate in a tree:



- A **Stock** is the named material (paper, board, plastic film). It has a name, description, vendor, notes, an optional external ID, and a stock type.
- A **Grade** is a specific weight/caliper combination on a stock. A stock can have many grades.
- A **Sheet** is a cut-to-size piece that belongs to a grade. Sheets have a width × height and a grain direction.
- A **Roll** is a continuous-width material that belongs to a grade. Rolls have a width (no fixed length) and a grain direction.

Every action you can take in Phoenix Stocks targets exactly one of these four levels.

3.2 The selector pattern (By / Operator / Value)

To pick which item(s) an action operates on, the form uses three fields that always come together:

Field	What it does
By	The field on the target item to look at. One of: ID, Name, External ID, Description, Notes.
Operator	How to compare. One of: Equals, Not equals, Starts with, Does not start with, Contains, Does not contain, Matches (regex), Does not match.
Value	The text to compare against. Switch variables and script expressions are allowed.

Example: **By = Name, Operator = Starts with, Value = Hammermill** selects every stock whose name begins with "Hammermill".

Parent-stock and parent-grade filters

Anything below a stock (grades, sheets, rolls) lives inside a parent. To narrow the search, the form shows **two** extra selector triples when relevant:

- **Stock By / Operator / Value** — filters the parent stock. Appears for any action that targets a grade, sheet, or roll.
- **Grade By / Operator / Value** — filters the parent grade. Appears for any action that targets a sheet or roll.

For **Find / Check for / Edit / Delete**, leaving the parent **Value** blank means "**any parent**" — the action searches across all stocks (and all grades). For **Create**, a blank parent **Value** is **not** allowed: the element needs to know exactly which stock (and grade) to attach the new item to. If you leave it blank on a Create, the job will fail with a clear error.

3.3 Bulk operations

Edit, Delete, and Duplicate Stock act on **every** item the selector matches, not just the first one. If you write **By = Vendor, Operator = Equals, Value = OldVendor** and 12 stocks match, all 12 get the operation. This is by design, so you can build sweeping cleanup or rename flows in one element. If you want to act on a single specific item, use **By = ID** with the GUID — IDs are unique.

3.4 The "Override existing" toggle

Several Create actions (and Duplicate Stock) include an **Override existing** Yes/No field. When you set it to **Yes**, the element looks for an existing item that would collide with the one you're about to create — same name (for stocks) or same key dimension (for sheets/rolls/grades) — and **deletes it first** before creating the new one. When set to **No** (the default), the create will fail if a collision exists.

Whenever Override deletes something, the job is routed to the **Warning** output, so you can spot accidental data loss in your flow.

3.5 Connection routing summary

Every job comes out one of three connections:

Connection	When
Success	The action completed exactly as the user asked. The result JSON is attached as a child log.
Warning	The action completed, but something it did diverged from a literal reading of the input — usually Override deleted data, or "Edit existing" found nothing and the element created a new item instead. The result JSON is attached with warnings: [...] strings.
Error	Something failed. A Check for * action returning <i>no match</i> also routes here (that's how Check answers "no"). For genuine Phoenix API errors (HTTP 400/404), error JSON attached.

The per-action sections below list, where relevant, **exactly** which conditions cause that action to route to Warning.

4. Actions

The Action dropdown at the top of the property panel chooses one of 21 operations. Below each action, the **Properties** table lists every form field you'll see in Switch when that action is selected, in the order it appears in the panel. The **Result JSON** code block shows a realistic example of the file attached to the job on Success or Warning routing.

4.1 Find Stock

What it does. Returns every stock in your Phoenix library whose chosen field matches the selector. The result is a JSON array — empty if nothing matched. Find is read-only; nothing in Phoenix is changed.

Properties

Field	Type	Required	Default	Notes
By	Dropdown	Yes	Name	Field to filter on: ID, Name, External ID, Description, Notes.
Operator	Dropdown	Yes	Equals	Equals, Not equals, Starts with, Does not start with, Contains, Does not contain, Matches (regex), Does not match.
Value	Single-line	Yes	(blank)	Text to compare against. An empty Value works as a literal empty-string compare.

Result JSON

A JSON array of full stock records. Each element includes the stock's grades (each with its sheets and rolls embedded), so a single Find Stock gives you the complete tree.

```
[
  {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Hammermill 80lb Cover",
    "version": "3",
    "created-on": "2025-01-15T08:30:00Z",
    "modified-on": "2026-05-19T14:22:00Z",
    "description": "Premium coated cover stock",
    "notes": "Re-order through Domtar",
    "external-id": "SKU-12345",
    "stock-type": {
      "id": "coated-stock-001",
      "path": "/phoenix/libraries/v2/stock-types/coated-stock-001"
    },
    "vendor": "Hammermill",
    "path": "/phoenix/libraries/v2/stocks/550e8400-e29b-41d4-a716-446655440000",
    "grades": [
```

```

{
  "id": "grade-id-001",
  "name": "80lb Cover",
  "version": "1",
  "description": "80lb Cover",
  "notes": "",
  "external-id": "GRD-80C",
  "grade-display": "Weight",
  "weight": 80,
  "weight-units": "Lb",
  "weight-type": "Cover",
  "caliper": "0.010",
  "cost": 0.05,
  "cost-units": "PerSheet",
  "any-sheet-size": false,
  "sheets": [
    {
      "id": "sheet-001",
      "name": "Letter",
      "description": "Standard letter size",
      "notes": "",
      "external-id": "SHT-LETTER",
      "dimension1": "8.5in",
      "dimension2": "11in",
      "cost": 0,
      "cost-units": "FromGrade",
      "grain": "Long"
    }
  ],
  "rolls": [
    {
      "id": "roll-001",
      "name": "50in",
      "description": "50-inch wide roll",
      "notes": "",
      "external-id": "ROLL-50",
      "dimension1": "50in",
      "cost": 100.50,
      "cost-units": "PerMSF",
      "grain": "AlongRoll"
    }
  ]
}
]

```

If no stocks match, the result is [] (an empty array). The job still routes to Success — "no match" is a valid Find answer.

4.2 Find Grade

What it does. Returns every grade across your library (or within a stock you filter for) that matches the grade selector. The result is an array; each entry tells you which parent stock the grade belongs to.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	Parent-stock filter field.
Stock Operator	Dropdown	Yes	Equals	Parent-stock operator.
Stock Value	Single-line	No	(blank)	Leave blank to search across all stocks. Set to filter to one stock or many.
By	Dropdown	Yes	Name	Grade filter field: ID, Name, External ID, Description, Notes.
Operator	Dropdown	Yes	Equals	Grade operator (same list as above).
Value	Single-line	No	(blank)	Leave blank to match every grade inside the selected stocks.

Result JSON

```
[
  {
    "stockId": "550e8400-e29b-41d4-a716-446655440000",
    "stockName": "Hammermill 80lb Cover",
    "grade": {
      "id": "grade-id-001",
      "name": "80lb Cover",
      "version": "1",
      "description": "80lb Cover",
      "notes": "",
      "external-id": "GRD-80C",
      "grade-display": "Weight",
      "weight": 80,
      "weight-units": "Lb",
      "weight-type": "Cover",
      "caliper": "0.010",
      "cost": 0.05,
      "cost-units": "PerSheet",
      "any-sheet-size": false,
      "sheets": [/* ... */],
      "rolls": [/* ... */]
    }
  }
]
```

4.3 Find Sheet

What it does. Returns every sheet matching the selector, with the parent stock and parent grade attached for context. The result is an array.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	Parent-stock filter field.
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	(blank)	Blank = all stocks.
Grade By	Dropdown	Yes	Name	Parent-grade filter field.
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	No	(blank)	Blank = all grades inside the selected stocks.
By	Dropdown	Yes	Name	Sheet filter field.
Operator	Dropdown	Yes	Equals	
Value	Single-line	No	(blank)	Blank = every sheet within the selected grades.

Result JSON

```
[
  {
    "stockId": "550e8400-e29b-41d4-a716-446655440000",
    "stockName": "Hammermill 80lb Cover",
    "gradeId": "grade-id-001",
    "gradeName": "80lb Cover",
    "item": {
      "id": "sheet-001",
      "name": "Letter",
      "description": "Standard letter size",
      "notes": "",
      "external-id": "SHT-LETTER",
      "dimension1": "8.5in",
      "dimension2": "11in",
      "cost": 0,
      "cost-units": "FromGrade",
      "grain": "Long"
    }
  }
]
```

4.4 Find Roll

What it does. Returns every roll matching the selector, with parent stock and parent grade for context. The result is an array.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	(blank)	Blank = all stocks.
Grade By	Dropdown	Yes	Name	
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	No	(blank)	Blank = all grades.
By	Dropdown	Yes	Name	Roll filter field.
Operator	Dropdown	Yes	Equals	
Value	Single-line	No	(blank)	Blank = every roll within the selected grades.

Result JSON

```
[
  {
    "stockId": "550e8400-e29b-41d4-a716-446655440000",
    "stockName": "Hammermill 80lb Cover",
    "gradeId": "grade-id-001",
    "gradeName": "80lb Cover",
    "item": {
      "id": "roll-001",
      "name": "50in",
      "description": "50-inch wide roll",
      "notes": "",
      "external-id": "ROLL-50",
      "dimension1": "50in",
      "cost": 100.50,
      "cost-units": "PerMSF",
      "grain": "AlongRoll"
    }
  }
]
```

4.5 Check for Stock

What it does. Tells you, with a Yes/No outcome, whether any stock matches the selector. Behind the scenes it runs the same logic as Find Stock and asks "did I get at least one result?"

- **Match found** → job goes out the **Success** port. **No result file is written**; the answer is logged inline to the job log.
- **No match** → job goes out the **Error** port. This is by design — it makes "is this stock missing?" trivial to branch on in a flow.
-

Properties

Field	Type	Required	Default	Notes
By	Dropdown	Yes	Name	
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	<i>(blank)</i>	

Result JSON

None. Check actions don't attach a result file. Check the job log for a message like:

```
Check for Stock: 3 match(es) — true.
```

4.6 Check for Grade

Same Yes/No outcome as Check for Stock, but at the grade level — searches grades (optionally narrowed by parent stock) and returns Success when at least one matches, Error when none do.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	<i>(blank)</i>	Blank = all stocks.
By	Dropdown	Yes	Name	
Operator	Dropdown	Yes	Equals	
Value	Single-line	No	<i>(blank)</i>	Blank = any grade inside the selected stocks.

Result JSON

None. (See [Check for Stock.](#))

4.7 Check for Sheet

Same Yes/No outcome, narrowed by parent stock and parent grade.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	(blank)	Blank = all stocks.
Grade By	Dropdown	Yes	Name	
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	No	(blank)	Blank = all grades.
By	Dropdown	Yes	Name	
Operator	Dropdown	Yes	Equals	
Value	Single-line	No	(blank)	Blank = any sheet.

Result JSON

None.

4.8 Check for Roll

Same Yes/No outcome, but for rolls.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	<i>(blank)</i>	Blank = all stocks.
Grade By	Dropdown	Yes	Name	
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	No	<i>(blank)</i>	Blank = all grades.
By	Dropdown	Yes	Name	
Operator	Dropdown	Yes	Equals	
Value	Single-line	No	<i>(blank)</i>	Blank = any roll.

Result JSON

None.

4.9 Create Stock

What it does. Creates a brand-new stock in Phoenix. Optionally, in the same job, you can attach up to **three grades** to the new stock — and each of those grades can carry up to two sheets and two rolls of its own. This makes it possible to onboard a complete material (stock → grade → sheets/rolls) in a single Phoenix Stocks element.

If you set **Override existing** to Yes and a stock with that name already exists, the existing stock is deleted first.

Properties — the stock itself

Field	Type	Required	Default	Notes
Name	Single-line	Yes	(blank)	The new stock's name. Example: Hammermill 80lb Cover.
Override existing	Yes/No	No	No	If Yes, delete any existing stock with this name before creating. Routes to Warning when something is actually deleted.
Description	Single-line	No	(blank)	
Notes	Single-line	No	(blank)	
External ID	Single-line	No	(blank)	For cross-referencing with an MIS or ERP.
Vendor	Single-line	No	(blank)	
Stock Type	Dropdown	No	Paper	Picked by name from Phoenix's stock-types library (populated live from your Phoenix server). Leave blank to omit.

Properties — Grade 1 (optional)

Toggle **Grade 1** to Yes to add a grade to the new stock. Set **Edit existing** to Yes if (in a re-run) you want the element to look up an already-attached grade and update it instead of creating a duplicate.

Field	Type	Required	Default	Notes
Grade 1	Yes/No	No	No	Master toggle for everything below.
Edit existing	Yes/No	No	No	If Yes, the By/Operator/Value below are used to find an existing grade to update.
By	Dropdown	Yes when Edit existing = Yes	Name	
Operator	Dropdown	Yes when Edit existing = Yes	Equals	
Value	Single-line	Yes when Edit existing = Yes	(blank)	Value to match the existing grade on.

Field	Type	Required	Default	Notes
Grade Display	Dropdown	Yes when Grade 1 = Yes	Weight	Weight, Caliper, or Both. Controls which of Weight/Caliper are required below.
Weight	Single-line	Yes when Grade Display is Weight or Both	0	Numeric. Example: 80.
Units	Dropdown	Yes when Weight is set	GSM	GSM or Lb.
Type	Dropdown	Yes when Weight is set	Text	Text, Book, Bond, Offset, Cover, Bristol, Index, Tag, Card.
Caliper	Single-line	Yes when Grade Display is Caliper or Both	0.0 mm	Thickness. Example: 0.010.
Any Sheet Size	Yes/No	Yes when Grade 1 = Yes	No	If Yes, the grade isn't limited to its defined sheet sizes.
Cost	Single-line	Yes when Grade 1 = Yes	0	Numeric.
Cost Units	Dropdown	Yes when Cost is set	FromGrade	See "Cost units" below.
Description	Single-line	No	(blank)	
Notes	Single-line	No	(blank)	
External ID	Single-line	No	(blank)	
Override existing	Yes/No	No	No	If Yes, deletes any existing grade on the stock with the same weight before adding. (This toggle is shown on Edit Stock as well but is ignored there.)

Properties — Grade 1 → Sheet 1 (optional)

Toggle **Sheet 1** under Grade 1 to attach a sheet to the new grade.

Field	Type	Required	Default	Notes
Sheet 1	Yes/No	No	No	Master toggle for the sheet's fields.
Edit existing	Yes/No	No	No	(Relevant only when Grade 1 has Edit existing = Yes too.)
By	Dropdown	Yes when Edit existing = Yes	Name	
Operator	Dropdown	Yes when Edit existing = Yes	Equals	
Value	Single-line	Yes when Edit existing = Yes	(blank)	

Field	Type	Required	Default	Notes
Width	Single-line	Yes when Sheet 1 = Yes	(blank)	Example: 8.5in or 216mm.
Height	Single-line	Yes when Sheet 1 = Yes	(blank)	Example: 11in or 279mm.
Grain	Dropdown	Yes when Sheet 1 = Yes	Long	Long, Short, None.
Cost	Single-line	Yes when Sheet 1 = Yes	0	
Cost Units	Dropdown	Yes when Cost is set	FromGrade	See "Cost units" below.
Description	Single-line	No	(blank)	
Notes	Single-line	No	(blank)	
External ID	Single-line	No	(blank)	

Properties — Grade 1 → Sheet 2 (optional)

Identical to Sheet 1 above. Toggle **Sheet 2** under Grade 1 to add a second sheet.

Properties — Grade 1 → Roll 1 (optional)

Toggle **Roll 1** under Grade 1 to attach a roll.

Field	Type	Required	Default	Notes
Roll 1	Yes/No	No	No	Master toggle for the roll's fields.
Edit existing	Yes/No	No	No	
By	Dropdown	Yes when Edit existing = Yes	Name	
Operator	Dropdown	Yes when Edit existing = Yes	Equals	
Value	Single-line	Yes when Edit existing = Yes	(blank)	
Width	Single-line	Yes when Roll 1 = Yes	(blank)	Example: 50in or 1270mm.
Grain	Dropdown	Yes when Roll 1 = Yes	AlongRoll	AcrossRoll, AlongRoll, None.
Cost	Single-line	Yes when Roll 1 = Yes	0	
Cost Units	Dropdown	Yes when Cost is set	FromGrade	
Description	Single-line	No	(blank)	
Notes	Single-line	No	(blank)	

Field	Type	Required	Default	Notes
External ID	Single-line	No	<i>(blank)</i>	

Properties — Grade 1 → Roll 2 (optional)

Identical to Roll 1 above.

Properties — Grade 2 and Grade 3 (optional)

Toggle **Grade 2** or **Grade 3** to add a second or third grade to the new stock. Each grade has the same fields as Grade 1 above — including its own optional Sheet 1, Sheet 2, Roll 1, Roll 2 sub-sections.

Cost units

The Cost Units dropdown (used by every grade, sheet, and roll Cost field) accepts:

- FromGrade
- PerSheet
- Per500Sheets
- Per1000Sheets
- PerLb
- Per500Lb
- Per1000Lb
- PerKg
- Per1000Kg
- PerFt2
- PerM2
- PerFt
- PerM
- PerMSF
- PerMSI.

FromGrade (the default for sheets and rolls) means "inherit from the parent grade".

Result JSON

```
{
  "action": "Create Stock",
  "name": "Hammermill 80lb Cover",
  "newStockId": "550e8400-e29b-41d4-a716-446655440001",
  "overrideDeletedId": "550e8400-e29b-41d4-a716-446655440099",
  "inlineGradesAttached": 2,
  "response": {
    "success": true,
    "status-code": 201,
    "errors": [],
    "warnings": [],
    "resources": [
      "/phoenix/libraries/v2/stocks/550e8400-e29b-41d4-a716-446655440001"
    ]
  },
  "inlineGradesResponse": {
    "success": true,
    "status-code": 200,
    "errors": [],
    "warnings": [],
    "resources": [
      "/phoenix/libraries/v2/stocks/550e8400-e29b-41d4-a716-446655440001"
    ]
  },
  "warnings": [
    "Create Stock: Override deleted existing stock \"Hammermill 80lb Cover\" (550e8400-e29b-41d4-a716-446655440099) before creating the new one."
  ]
}
```

Notes on fields:

- `newStockId` — GUID of the newly created stock. Empty string if Phoenix's response didn't include a URI.
- `overrideDeletedId` — only present when Override actually deleted a pre-existing stock. Absent otherwise.
- `inlineGradesAttached` — count of inline grades the element added in this run (0–3).
- `inlineGradesResponse` — Phoenix's API response from the second call that wrote the grades. null if no inline grades were active.
- `response` — Phoenix's API response from the initial create.
- `warnings` — see the bullets below.

When this action emits a Warning

- **Override existing = Yes** actually deleted a pre-existing stock with the same name.
- An inline-grade's **Edit existing = Yes** matched no existing grade on the new stock, so the element added it as a brand-new grade.
- An inline-grade's weight matches an existing grade and that grade's **Override existing = No** — the new grade is skipped and the existing one is reused.
- An inline-grade's **Override existing = Yes** actually deleted one or more existing same-weight grades.
- A rare programming-defence case: the new stock's ID couldn't be parsed from Phoenix's response, so the inline-grade attachment was skipped.

4.10 Create Grade

What it does. Adds a new grade to one or more existing stocks. The parent stock selector is **required** (you must say which stock to attach to). Bulk: if the parent selector matches multiple stocks, the grade is added to every one. Optionally, the new grade can carry up to two sheets and two rolls created in the same call.

Same-weight protection. Phoenix's GUI treats weight as a unique key per stock — having two grades with the same weight on one stock is messy. Phoenix Stocks mirrors that: if a stock already has a grade with the requested weight, the element will either reuse the existing grade (Override = No, the default) or delete the duplicates first (Override = Yes). In both cases the job routes to **Warning** so you know it happened.

Properties — selector for the parent stock

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	Yes	<i>(blank)</i>	Required. Blank is rejected for Create Grade.

Properties — the new grade

Field	Type	Required	Default	Notes
Grade Display	Dropdown	Yes	Weight	Weight, Caliper, Both.
Weight	Single-line	Yes when Grade Display is Weight or Both	0	Numeric.
Units	Dropdown	Yes when Weight is set	GSM	GSM or Lb.
Type	Dropdown	Yes when Weight is set	Text	Text, Book, Bond, Offset, Cover, Bristol, Index, Tag, Card.
Caliper	Single-line	Yes when Grade Display is Caliper or Both	0.0 mm	Thickness.
Any Sheet Size	Yes/No	No	No	
Cost	Single-line	No	0	
Cost Units	Dropdown	Yes when Cost is set	FromGrade	See "Cost units" in section 4.9.
Description	Single-line	No	<i>(blank)</i>	
Notes	Single-line	No	<i>(blank)</i>	

Field	Type	Required	Default	Notes
External ID	Single-line	No	<i>(blank)</i>	
Override existing	Yes/No	No	No	If Yes, delete any same-weight grade on each matched stock before adding.

Properties — Sheet 1 (optional) / Sheet 2 (optional)

Same nested structure as in [Create Stock](#) Grade 1 → Sheet 1. Toggle **Sheet 1** to add a sheet to the new grade; Sheet 2 to add a second.

Properties — Roll 1 (optional) / Roll 2 (optional)

Same nested structure as in [Create Stock](#) Grade 1 → Roll 1.

Result JSON

```
{
  "action": "Create Grade",
  "weight": 80,
  "weightUnits": "Lb",
  "caliper": "0.010",
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    }
  ],
  ...
},
"warnings": [
  "Create Grade: weight=80 already exists in stock \"Cream Wove\" (existing id grade-id-099); skipped because Override is off — using the existing grade."
]
}
```

Notes on per-stock entries:

- A normal create produces an entry with a response (Phoenix's API response).
- A same-weight-skip produces an entry with skipped: true and existingGradeId (no response).

When this action emits a Warning

- A stock already had a same-weight grade and **Override existing = No** — the new grade was skipped and the existing one is reused.
- **Override existing = Yes** actually deleted one or more same-weight grades on a stock before adding.

4.11 Create Sheet

What it does. Adds a new sheet to one or more grades inside one or more stocks. Both the parent stock and parent grade selectors are **required**. Bulk: the same sheet is added to every matched grade.

If you set **Override existing** to Yes and any matched grade already has a sheet with the same width × height, those are deleted before the new sheet is added.

Properties — parent selectors

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	Yes	<i>(blank)</i>	Required for Create Sheet.
Grade By	Dropdown	Yes	Name	
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	Yes	<i>(blank)</i>	Required for Create Sheet.

Properties — the new sheet

Field	Type	Required	Default	Notes
Width	Single-line	Yes	<i>(blank)</i>	Example: 8.5in or 216mm.
Height	Single-line	Yes	<i>(blank)</i>	Example: 11in or 279mm.
Grain	Dropdown	Yes	Long	Long, Short, None.
Cost	Single-line	Yes	0	
Cost Units	Dropdown	Yes when Cost is set	FromGrade	
Description	Single-line	No	<i>(blank)</i>	
Notes	Single-line	No	<i>(blank)</i>	
External ID	Single-line	No	<i>(blank)</i>	
Override existing	Yes/No	No	No	If Yes, deletes same-dimensions sheets in each matched grade before adding.

Result JSON

```
{
  "action": "Create Sheet",
  "dim1": "8.5in",
  "dim2": "11in",
  "totalGradesTouched": 3,
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "gradesTouched": 2,
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    },
    ...
  ],
  "warnings": [
    "Create Sheet: Override removed 1 existing sheet(s) with dim1=8.5in, dim2=11in from grade \"80lb Cover\" in stock \"Hammermill 80lb Cover\" before adding the new sheet."
  ]
}
```

When this action emits a Warning

- **Override existing = Yes** actually deleted one or more same-dimensions sheets in any matched grade.

4.12 Create Roll

What it does. Adds a new roll to one or more grades inside one or more stocks. Both parent selectors are **required**. Bulk: the same roll is added to every matched grade.

If you set **Override existing** to Yes, rolls with the same width on any matched grade are deleted first.

Properties — parent selectors

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	Yes	(blank)	Required.
Grade By	Dropdown	Yes	Name	
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	Yes	(blank)	Required.

Properties — the new roll

Field	Type	Required	Default	Notes
Width	Single-line	Yes	(blank)	Example: 50in or 1270mm.
Grain	Dropdown	Yes	AlongRoll	AcrossRoll, AlongRoll, None.
Cost	Single-line	Yes	0	
Cost Units	Dropdown	Yes when Cost is set	FromGrade	
Description	Single-line	No	(blank)	
Notes	Single-line	No	(blank)	
External ID	Single-line	No	(blank)	
Override existing	Yes/No	No	No	If Yes, deletes same-width rolls in each matched grade before adding.

Result JSON

```
{
  "action": "Create Roll",
  "dim1": "50in",
  "totalGradesTouched": 2,
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "gradesTouched": 1,
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    }
  ],
  "warnings": [
    "Create Roll: Override removed 2 existing roll(s) with dim1=50in from grade \"80lb Cover\" in stock \"Hammermill 80lb Cover\" before adding the new roll."
  ]
}
```

When this action emits a Warning

- **Override existing = Yes** actually deleted one or more same-width rolls in any matched grade.

4.13 Edit Stock

What it does. Modifies one or more existing stocks selected by the By/Operator/Value triple. Any of the stock's top-level fields (name, description, notes, external ID, vendor, stock type) can be updated; leave a field blank to keep the existing value. Optionally, the same call can add or update up to three inline grades on each matched stock.

Bulk: if your selector matches 5 stocks, the same edit is applied to all 5. Be careful with Name — setting a new Name on a selector that matches many stocks will leave you with multiple stocks all named the same thing.

Properties — selector

Field	Type	Required	Default	Notes
By	Dropdown	Yes	Name	Field to identify which stock(s) to edit.
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	<i>(blank)</i>	Required — the element will refuse to run if you leave this blank, because a blank-equals-anything edit is almost never what you want.

Properties — stock fields

Field	Type	Required	Default	Notes
Name	Single-line	No	<i>(blank)</i>	New name. Blank = keep existing.
Description	Single-line	No	<i>(blank)</i>	Blank = keep existing.
Notes	Single-line	No	<i>(blank)</i>	Blank = keep existing.
External ID	Single-line	No	<i>(blank)</i>	Blank = keep existing.
Vendor	Single-line	No	<i>(blank)</i>	Blank = keep existing.
Stock Type	Dropdown	No	Paper	Blank = keep existing.

Properties — Grade 1 / Grade 2 / Grade 3 (optional)

Identical to the Grade sub-sections in [Create Stock](#). Toggle a grade on to either add a new grade to the matched stocks or — with **Edit existing = Yes** — to find and update a grade that's already there. The grade's own **Override existing** toggle is shown but is **ignored on Edit Stock** (it's only honoured on Create Stock).

Result JSON

```
{
  "action": "Edit Stock",
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill Premium 80lb Cover",
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    }
  ],
  "warnings": [
    "Edit Stock: inline-grade \"edit existing\" matched no grade with Name=\"OldGrade\" in stock  
\"Hammermill Premium 80lb Cover\" — added as new grade instead."
  ]
}
```

The `name` field reflects the **updated** name after the edit.

When this action emits a Warning

- An inline-grade's **Edit existing = Yes** matched no grade and the element added it as new.
- An inline-grade's weight matches an existing grade and that inline-grade's **Override existing = No** — the new grade is skipped and the existing one is reused.
- An inline-grade's **Override existing = Yes** actually deleted one or more same-weight grades before adding.

4.14 Edit Grade

What it does. Modifies one or more grades, optionally narrowed by parent stock. Any field left blank keeps the existing value. Optionally, the same call can add or update standalone sheets and rolls on each matched grade.

Bulk: the same edit is applied to every matched grade across every matched stock.

Properties — parent and target selectors

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	(blank)	Blank = search across all stocks.
By	Dropdown	Yes	Name	Selects which grade(s) to edit.
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	(blank)	Required.

Properties — grade fields

Same list as [Create Grade](#)'s "the new grade" section. Blank fields keep the existing value:

- Weight applies only when greater than 0 (so the default 0 means "keep existing").
- Caliper applies only when non-blank.
- Cost applies only when greater than 0.
- Description, Notes, External ID, Weight Units, Weight Type, Cost Units, Grade Display — all apply only when non-blank.
- Any Sheet Size — its Yes/No value is always written.

Properties — Sheet 1 / Sheet 2 (optional) and Roll 1 / Roll 2 (optional)

Same nested structure as [Create Grade](#). Toggle a sheet or roll on to add a new one to the matched grades, or set **Edit existing = Yes** to update an existing sheet/roll inside each matched grade. When activeSheets or activeRolls are present, they **replace** the grade's sheets/rolls list — leave them off to keep existing sheets/rolls untouched.

Result JSON

```
{
  "action": "Edit Grade",
  "totalGradesTouched": 3,
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "gradesTouched": 2,
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    },
    ...
  ],
  "warnings": [
    "Edit Grade: standalone sheet \"edit existing\" matched no sheet with Name=\"Letter\" (grade id grade-id-001) in stock \"Hammermill 80lb Cover\" — added as new sheet instead."
  ]
}
```

When this action emits a Warning

- A standalone sheet's or roll's **Edit existing = Yes** matched no sheet/roll on a matched grade — the element added it as new.

4.15 Edit Sheet

What it does. Modifies one or more sheets, optionally narrowed by parent stock and parent grade. Any field left blank (or zero, for Cost) keeps the existing value. Bulk: the same edit is applied to every matched sheet.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	(blank)	Blank = all stocks.
Grade By	Dropdown	Yes	Name	
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	No	(blank)	Blank = all grades.
By	Dropdown	Yes	Name	Selects which sheet(s) to edit.
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	(blank)	
Width	Single-line	No	(blank)	Blank = keep existing.
Height	Single-line	No	(blank)	Blank = keep existing.
Grain	Dropdown	Yes	Long	Long, Short, None.
Cost	Single-line	No	0	0 = keep existing.
Cost Units	Dropdown	Yes when Cost is set	FromGrade	
Description	Single-line	No	(blank)	Blank = keep existing.
Notes	Single-line	No	(blank)	Blank = keep existing.
External ID	Single-line	No	(blank)	Blank = keep existing.

Result JSON

```
{
  "action": "Edit Sheet",
  "totalSheetsTouched": 2,
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "sheetsTouched": 2,
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    }
  ]
}
```

Edit Sheet does not emit Warnings — it always routes Success (or Error on API failure).

4.16 Edit Roll

What it does. Modifies one or more rolls, optionally narrowed by parent stock and parent grade. Any field left blank (or zero, for Cost) keeps the existing value. Bulk: the same edit is applied to every matched roll.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	(blank)	Blank = all stocks.
Grade By	Dropdown	Yes	Name	
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	No	(blank)	Blank = all grades.
By	Dropdown	Yes	Name	Selects which roll(s) to edit.
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	(blank)	
Width	Single-line	No	(blank)	Blank = keep existing.
Grain	Dropdown	Yes	AlongRoll	AcrossRoll, AlongRoll, None.
Cost	Single-line	No	0	0 = keep existing.
Cost Units	Dropdown	Yes when Cost is set	FromGrade	
Description	Single-line	No	(blank)	Blank = keep existing.
Notes	Single-line	No	(blank)	Blank = keep existing.
External ID	Single-line	No	(blank)	Blank = keep existing.

Result JSON

```
{
  "action": "Edit Roll",
  "totalRollsTouched": 1,
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "rollsTouched": 1,
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    }
  ]
}
```

Edit Roll does not emit Warnings.

4.17 Duplicate Stock

What it does. Clones one or more existing stocks into new stocks with a new name (and optional new external ID, description, notes, vendor). The source is selected by the By/Operator/Value triple — and yes, the source can be a single stock or many.

Bulk and the ``-N`` suffix. If the source selector matches **1** stock, the clone gets exactly the **New Name** you typed. If it matches **N > 1** stocks, the element appends -1, -2, ... -N to the New Name (and to the New External ID if you provided one) to avoid name collisions across the clone set. So selecting 3 stocks with New Name = Master Stock yields three clones named Master Stock-1, Master Stock-2, Master Stock-3. When this happens the job routes to **Warning** so you know suffixes were applied.

What is and isn't copied. The clone is built from the source's writable fields only — grades, sheets and rolls are cloned in full, but every read-only field (IDs, created-on/modified-on timestamps, version numbers) is freshly assigned by Phoenix. **External IDs on cloned grades, sheets, and rolls are always cleared** — external IDs are unique-key references to upstream systems and shouldn't collide between source and clone. Top-level overrides come from the New ... fields below: New Description / Notes / Vendor are taken from the source if you leave them blank, or overridden uniformly across the clone set if you fill them in. New External ID is **cleared on the clone if you leave it blank** (since it shouldn't collide); set it to give every clone the same external ID (suffixed when N > 1).

Override existing. If you set Override = Yes and a stock with the resolved clone name already exists (and isn't itself one of the source stocks), that stock is deleted before each clone is created.

Properties — source selector

Field	Type	Required	Default	Notes
By	Dropdown	Yes	Name	Identifies the source stock(s) to clone.
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	(blank)	Required.

Properties — clone target

Field	Type	Required	Default	Notes
New Name	Single-line	Yes	(blank)	Required. Will get -1/-2/... appended when the source selector matches more than one stock.
New External ID	Single-line	No	(blank)	If blank, every clone's external ID is cleared. If set, it's applied to every clone (suffixed when the source matches more than one stock).
New Description	Single-line	No	(blank)	Blank = copy each clone's description from its source. Non-blank = override on every clone.

Field	Type	Required	Default	Notes
New Notes	Single-line	No	(blank)	Blank = copy from source. Non-blank = override.
New Vendor	Single-line	No	(blank)	Blank = copy from source. Non-blank = override.
Override existing	Yes/No	No	No	If Yes, delete any pre-existing stock with the resolved clone name (post-suffix) before each clone. Will refuse to delete a stock that's currently in the source set.

Result JSON (N = 1)

```
{
  "action": "Duplicate Stock",
  "sourceMatchCount": 1,
  "suffixed": false,
  "stocksAffected": [
    {
      "sourceStockId": "stock-001",
      "sourceStockName": "Hammermill 80lb Cover",
      "newStockId": "550e8400-e29b-41d4-a716-446655440042",
      "newStockName": "Hammermill 80lb Cover (Copy)",
      "newExternalId": "SKU-12345-COPY",
      "gradesCloned": 3,
      "sheetsCloned": 9,
      "rollsCloned": 3,
      "response": {
        "success": true,
        "status-code": 201,
        "errors": [],
        "warnings": [],
        "resources": [
          "/phoenix/libraries/v2/stocks/550e8400-e29b-41d4-a716-446655440042"
        ]
      }
    }
  ],
  "warnings": []
}
```

Result JSON (N > 1, with suffixes)

```
{
  "action": "Duplicate Stock",
  "sourceMatchCount": 3,
  "suffixed": true,
  "stocksAffected": [
    {
      "sourceStockId": "stock-001",
      "sourceStockName": "Hammermill 80lb Cover",
      "newStockId": "550e8400-e29b-41d4-a716-446655440043",
      "newStockName": "Master Stock-1",
      "newExternalId": "EXT-MASTER-1",
      "gradesCloned": 3,
      "sheetsCloned": 9,
      "rollsCloned": 3,
      "response": { "success": true, "status-code": 201, "errors": [], "warnings": [], "resources": [
"/phoenix/libraries/v2/stocks/550e8400-e29b-41d4-a716-446655440043" ] }
    },
    {
      "sourceStockId": "stock-002",
      "sourceStockName": "Cream Wove",
      "newStockId": "550e8400-e29b-41d4-a716-446655440044",
      "newStockName": "Master Stock-2",
      "newExternalId": "EXT-MASTER-2",
      "gradesCloned": 2,
      "sheetsCloned": 6,
      "rollsCloned": 2,
      "response": { "success": true, "status-code": 201, "errors": [], "warnings": [], "resources": [
"/phoenix/libraries/v2/stocks/550e8400-e29b-41d4-a716-446655440044" ] }
    },
  ],
  "warnings": [
    "Duplicate Stock: source selector matched 3 stocks — \"-1\", \"-2\", ... indexed suffixes were
appended to the New Name and to the New External ID to avoid collisions across the clone set.",
    "Duplicate Stock: Override deleted existing stock \"Master Stock-3\" (550e8400-e29b-41d4-a716-
446655440099) before cloning into that name."
  ]
}
```

Notes on per-clone entries:

- `sourceStockId` / `sourceStockName` — the source the clone was made from.
- `newStockId` / `newStockName` / `newExternalId` — the resolved names/IDs on the new stock (after any -N suffix).
- `overrideDeletedId` — only present when Override actually deleted a pre-existing stock for this clone.
- `gradesCloned` / `sheetsCloned` / `rollsCloned` — counts of nested entities cloned from the source.

When this action emits a Warning

- The source selector matched more than one stock and the element applied -1, -2, ... suffixes.
- **Override existing = Yes** actually deleted one or more pre-existing stocks for any of the clones.

4.18 Delete Stock

What it does. Deletes one or more stocks selected by the By/Operator/Value triple. Bulk: every matched stock is deleted. There is no Warning routing for Delete Stock — deletion is the point, so there's nothing to flag as a divergence.

Properties

Field	Type	Required	Default	Notes
By	Dropdown	Yes	Name	
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	(blank)	Required.

Result JSON

```
{
  "action": "Delete Stock",
  "deletedStocks": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "response": {
        "success": true,
        "status-code": 204,
        "errors": [],
        "warnings": [],
        "resources": []
      }
    },
    {
      "id": "stock-002",
      "name": "Cream Wove",
      "response": {
        "success": true,
        "status-code": 204,
        "errors": [],
        "warnings": [],
        "resources": []
      }
    }
  ]
}
```

The response field may be null in the rare case where Phoenix returns no body (e.g. the stock was already gone by the time the DELETE call landed). Delete Stock does not include a warnings field at all.

4.19 Delete Grade

What it does. Deletes one or more grades from one or more stocks. Bulk: every matched grade across every matched stock is removed.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	(blank)	Blank = all stocks.
By	Dropdown	Yes	Name	Selects which grade(s) to delete.
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	(blank)	Required.

Result JSON

```
{
  "action": "Delete Grade",
  "totalItemsRemoved": 2,
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "itemsRemoved": 1,
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    },
    ...
  ]
}
```

If no grades matched in any of the matched stocks, the action fails with a clear error and the job routes to Error. Delete Grade does not currently emit Warnings.

4.20 Delete Sheet

What it does. Deletes one or more sheets from one or more grades inside one or more stocks. Bulk.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	(blank)	Blank = all stocks.
Grade By	Dropdown	Yes	Name	
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	No	(blank)	Blank = all grades.
By	Dropdown	Yes	Name	Selects which sheet(s) to delete.
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	(blank)	Required.

Result JSON

```
{
  "action": "Delete Sheet",
  "totalItemsRemoved": 3,
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "itemsRemoved": 3,
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    }
  ]
}
```

If no sheets matched anywhere, the job fails to Error. Delete Sheet does not currently emit Warnings.

4.21 Delete Roll

What it does. Deletes one or more rolls from one or more grades inside one or more stocks. Bulk.

Properties

Field	Type	Required	Default	Notes
Stock By	Dropdown	Yes	Name	
Stock Operator	Dropdown	Yes	Equals	
Stock Value	Single-line	No	(blank)	Blank = all stocks.
Grade By	Dropdown	Yes	Name	
Grade Operator	Dropdown	Yes	Equals	
Grade Value	Single-line	No	(blank)	Blank = all grades.
By	Dropdown	Yes	Name	Selects which roll(s) to delete.
Operator	Dropdown	Yes	Equals	
Value	Single-line	Yes	(blank)	Required.

Result JSON

```
{
  "action": "Delete Roll",
  "totalItemsRemoved": 2,
  "stocksAffected": [
    {
      "id": "stock-001",
      "name": "Hammermill 80lb Cover",
      "itemsRemoved": 2,
      "response": {
        "success": true,
        "status-code": 200,
        "errors": [],
        "warnings": [],
        "resources": [ "/phoenix/libraries/v2/stocks/stock-001" ]
      }
    }
  ]
}
```

If no rolls matched anywhere, the job fails to Error. Delete Roll does not currently emit Warnings.

5. Output files

Every job that passes through PhoenixStocks may attach **one** JSON file to itself as a Switch child log. There are two possible files and which one (if any) appears depends on how the job left the element.

5.1 <jobName>-result.json

Written when the action completed (Success **or** Warning routing). The shape of this file is action-specific — every section above shows the exact structure for its action.

A few rules apply across all actions:

- For **Find** actions, the file is a JSON **array** at the top level.
- For **Create / Edit / Duplicate / Delete-Grade-or-Sheet-or-Roll** actions, the file is a JSON **object** with at minimum an **action** field naming the operation.
- **Create / Edit / Duplicate** actions also include a top-level **warnings: string[]** array. An empty array means the job went Success; a non-empty array means it went Warning, and each string explains one specific divergence in plain English.
- **Delete Stock** has no **warnings** field at all. **Delete Grade / Sheet / Roll, Edit Sheet, and Edit Roll** also omit the field (these actions don't currently have any Warning paths).
- For **Check** actions, no file is written. The Yes/No outcome drives Success vs Error routing only.

5.2 <jobName>-error.json

Written when Phoenix returned a structured API error (HTTP 400 / 404), or when 503 retries were exhausted (see [5.3](#) below). The file contains Phoenix's own **ResponseEntity** so you can parse the underlying problem in a downstream script.

```
{
  "success": false,
  "status-code": 404,
  "errors": [
    {
      "id": 404,
      "text": "Stock not found",
      "action": "GET /phoenix/libraries/v2/stocks/stock-missing"
    }
  ],
  "warnings": [],
  "resources": []
}
```

If Phoenix returned a body that wasn't valid JSON, the file instead contains a wrapper:

```
{
  "raw": "<the raw text Phoenix returned>",
  "status-code": 500
}
```

Errors that don't write a file. Some errors — bad configuration, no matches on an Edit/Delete selector, a missing required Value on Create — are validation errors from the element itself, not Phoenix. These route to Error and write a clear message to the **flow log** but do **not** attach an error JSON file.

5.3 503 retry behaviour

Phoenix reports "service at maximum concurrency" with HTTP 503 when it's overloaded. PhoenixStocks handles 503 automatically:

1. On the first 503, the element schedules a retry in **10 seconds** (via `job.processLater`) and the job goes back into the queue.
2. If 503 recurs, the next retry is at **15 seconds**, then **20**, then **25**.
3. After 5 total attempts all returning 503, the element gives up: it writes the error JSON, logs the failure, and routes the job to **Error**.

During retries no result or error file is written — the job is simply requeued. The retry count is stored in the job's private data under the key `503RetryCount`.

5.4 Concurrency safety

When multiple PhoenixStocks elements (in the same flow or in different flows) try to talk to Phoenix at the same time, they coordinate via a small on-disk lock so they don't trample each other's writes to Phoenix's library file. This is invisible in normal operation — you don't need to configure anything. The lock automatically recovers if a job crashes mid-operation.

6. Troubleshooting & FAQ

My job came out the Warning port — why?

Look in the result JSON (<jobName>-result.json) for the top-level warnings array. Each string explains, in plain language, one specific thing the element did that diverged from a literal reading of your input — almost always one of:

- Override existing was Yes and the element deleted something to make room for the new item.
- Edit existing was Yes but no matching item was found, so the element added it as a new item instead.
- A same-weight grade already existed on a stock and Override was Off, so the element reused the existing grade.
- Duplicate Stock matched more than one source stock, so -1, -2, ... was appended to your New Name.

The per-action sections above each list the full conditions for that action.

Switch says my Create Sheet (or Create Roll, or Create Grade) failed — and I left Stock Value blank

That's by design. Create actions need to know exactly which parent to attach to, so a blank Stock Value (or Grade Value, for Create Sheet/Roll) is rejected with an error. Set them to filter to a specific stock and grade.

My Edit Stock fails with "no stocks matched the selector"

Your By/Operator/Value combination didn't find anything in Phoenix. Double-check the field you're filtering on (e.g. is the stock actually named what you think?) and the operator (Equals is case-sensitive). Try first with a Find Stock using the same selector — if Find returns an empty array, Edit Stock will too.

Duplicate Stock added -1 and -2 to my stock names — what happened?

Your source selector matched more than one stock, so the element appended -1, -2, ... to your New Name (and to your New External ID, if you provided one) to avoid name collisions across the clone set. To get exactly the name you typed, narrow your source selector to a single stock — using By = ID with the source's GUID is the surest way.

Phoenix keeps reporting "at maximum concurrency"

That's HTTP 503. PhoenixStocks retries automatically up to 5 times with increasing delays. If you frequently exhaust the retries, your Phoenix server is being pushed harder than it can

handle — consider running flow-side throttling, or upgrading Phoenix's concurrency limits if your license allows it.

Where can I see the full Phoenix error message?

Two places:

1. The error JSON attached to the job: `<jobName>-error.json` under the job's child logs.
2. The flow log in Switch — the full Phoenix response body is logged on every API error.

Can I run multiple Phoenix Stocks elements at the same time against the same Phoenix server?

Yes. The element coordinates with other Phoenix Stocks elements on the same machine to avoid Phoenix file-write conflicts (see [5.4](#)). No configuration needed.

What's the difference between Check for and Find actions?

Find returns the full data (and an empty array [] if nothing matched, still routing Success). Check for only tells you Yes or No, and routes a "no" to the **Error** port — useful for branching like "if this stock doesn't exist, go create it."

Can I use Switch variables in the property fields?

Yes — every text field accepts Switch variables and inline script expressions, including the selector Value fields. This lets the element be fully data-driven from each job's metadata.

What if I want to act on exactly one specific stock?

Use `By = ID` with the stock's GUID as Value. IDs are unique across the library, so this always selects exactly one item (or zero, if the ID is wrong). All other By options (`Name`, `External ID`, `Description`, `Notes`) can match multiple items.