Shot Management

Classifying every shot point, signing a line off, and writing its clean SPS

SEG-D Monitor 2.31.0 · guide revision 2.31.0 · 2026-11-20

Contents

1. What shot management is **3**
1.1 Four answers for every shot point 3
1.2 Where the codes come from 3
1.3 What it changes, and what it does not 3

2. Setting it up at the start of a project **4**
2.1 The question when the project opens 4
2.2 Classes, codes and flags 5
2.3 The app's findings and their codes 5
2.4 Which sequence keeps an overlap 5

3. The routine, line by line **7**
3.1 Reviewing a line 7
3.2 Marking shot points 8
3.3 Signing off and the clean SPS 9

4. Situations, and what to do **10**

5. Where the figures show **12**
5.1 The line by line history 12
5.2 Days 12
5.3 The Survey, the report and the handover 12

6. What the app stops you doing **13**

7. Reference **14**
7.1 The standard scheme 14
7.2 The app's findings 14
7.3 The files 15

1. What shot management is

Shot management says what every shot point of the survey **counts as** for production: whether it goes into the line's clean SPS, whether it counts in the day's production, and whether the point is still to be shot. The app classifies every shot by itself from what the data shows; the crew adds what only the crew knows; and when a line's QC is finished the crew signs it off and writes its clean SPS. It is opened with 🎯 **Shots** in the Analysis group of the toolbar.

1.1 Four answers for every shot point

Every shot point carries one or more **codes**. Each code belongs to a **class**, and the class says what the shot point counts as. There are four answers, fixed in the app, because every figure depends on them; the classes and codes that give them are the project's own.

COUNTS AS	CLEAN SPS	DAY'S PRODUCTION	THE PREPLOT POINT	TYPICAL CLASSES
Production	In it	Counted	Done	Good, Minor, Marginal
Not counted	Not in it	Not counted	Not a hole - shot by another sequence, or off the plan	Overlap, Run-in / run-out
Reshoot	Not in it	Not counted - but the time it took is shooting time	Still to shoot	Bad / Skip, DNP / NTBP
Never to be shot	-	-	Leaves the plan	Never to be shot (client approval)

Every class and every code has a **flag**: a number, and the higher it is, the worse the shot point. A shot point with several codes takes its **worst** - its final code - and counts as that code's class says.

1.2 Where the codes come from

- **The app** - its findings, each mapped to a code: the Spec Check's bad shots (fire time, depth, pressure, source dead, nav mask, and time-break sync and firing spread where they are counted), a missed shot point (no SEG-D record), warnings (a reading outside its limit on fewer guns than make a bad shot, a gun not fired, an autofire), an **overlap** with another sequence of the same source line, a **run-in / run-out** shot off the preplot, and a line marked not to be processed. The bad shots are the Spec Check's own: one rule, its criteria as set in General Settings > Shot Points.
- **The crew** - *marks*: shot points of a line, a code and a comment, signed with a name and timed.
- **The plan** - *plan exclusions*: preplot points agreed with the client never to be shot.

1.3 What it changes, and what it does not

The **Spec Check is not changed**: its verdict on a line is still the app's own, from the data alone. Shot management classifies on top of it, with the crew's word. Once it is set up, the **Survey** counts production shots, the **End of Line report** gains a Shot Management page for each line, and the **shift handover** lists the changes made in the shift. Plan exclusions change one QC figure - an excluded point is never a missed shot - and that happens whether or not shot management is set up, because a point agreed never to be shot was never a shot to miss.

2. Setting it up at the start of a project

2.1 The question when the project opens

When a project is opened whose shot management is not set up, the app asks: **Set up now…**, **Later** (it asks again next time the project is opened) or **Don't ask for this project**. Until it is set up the window uses the standard scheme, and the Survey and the reports show what they always showed - nothing waits for it.

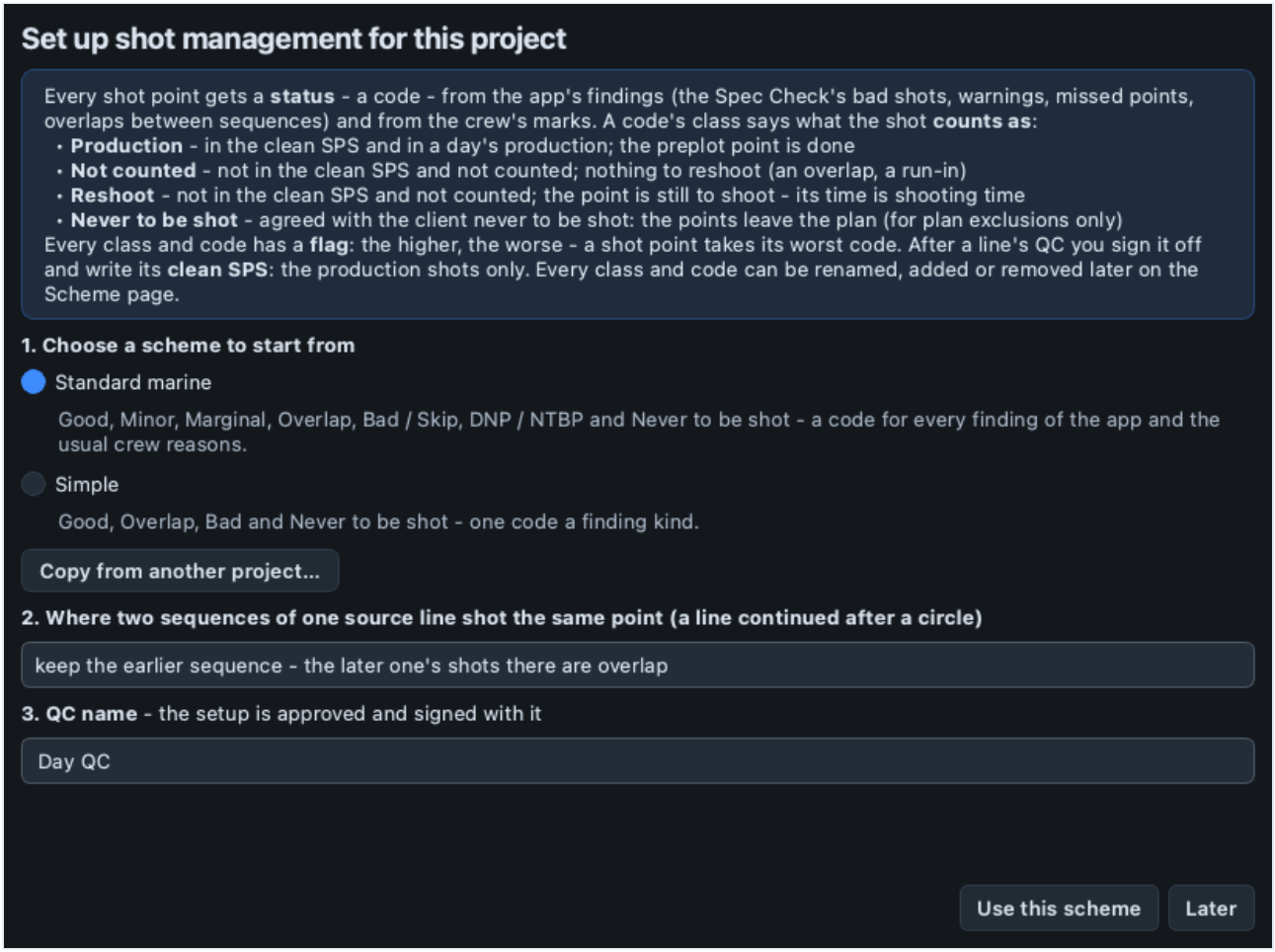

Figure 1. *Setting up: a scheme to start from, which sequence keeps a shot point two sequences produced, and the name the setup is signed with.*

To set shot management up

1. Choose **Standard marine** (a class for each kind of shot and a code for every finding and the usual crew reasons), **Simple** (Good, Overlap, Bad, Never to be shot), or **Copy from another project…** to take a scheme the crew already uses - the scheme only, never that project's marks.
2. Choose which sequence keeps a shot point two sequences produced (§2.4).
3. Type your name and press **Use this scheme**. The window opens on the **Scheme** page: adjust it to the contract (§2.2) and press **Save scheme**.

2.2 Classes, codes and flags

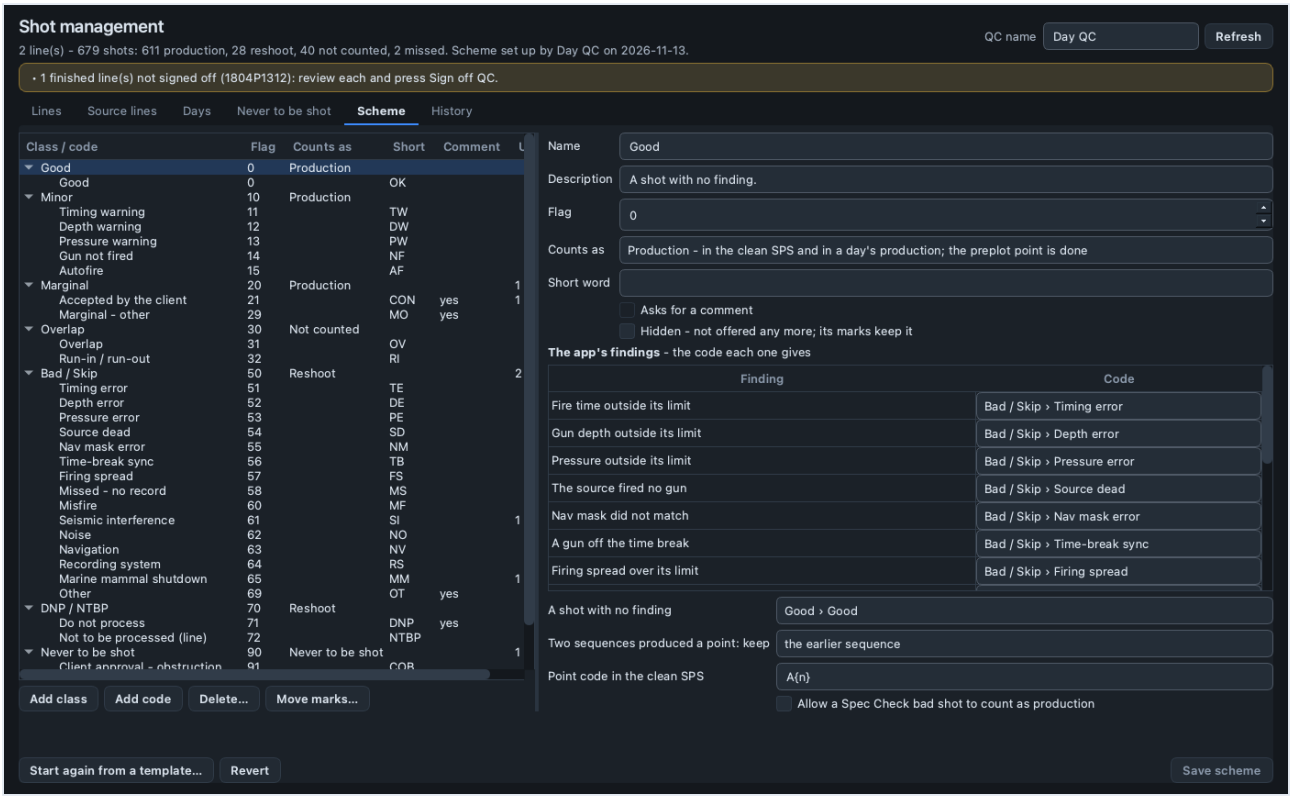

Figure 2. *The Scheme page: the classes and their codes with their flags, what each class counts as, the short word, whether a code asks for a comment and how many marks use it; on the right the item being edited, the code each finding of the app gives, and the options.*

Select a class or a code to edit its name, description, flag and - for a class - what it counts as; a code also has a **short word** (what the files carry) and can **ask for a comment**. **Add class**, **Add code**, **Delete…** (only what nothing uses) and **Hide** (no longer offered; its marks keep it). **Move marks…** moves every mark of a code to another, which is how a code in use is retired.

Flags and bands. A class's flag opens its band, and its codes lie inside it, below the next class's flag: with Minor at 10 and Marginal at 20, Minor's codes are 10 to 19. So comparing two codes' flags also compares their classes. **Production classes must have the lowest flags**: a code that takes a shot out of production then always wins over one that keeps it in.

> The scheme is checked as you edit it and **cannot be saved with a mistake**; the page says what is wrong and what to do. See chapter 6 for the full list.

2.3 The app's findings and their codes

Each finding of the app maps to a code (the list under *The app's findings*). A bad shot of the Spec Check normally maps to a reshoot code; mapping one to a production code is refused unless **Allow a Spec Check bad shot to count as production** is ticked - a contract decision, not a convenience. A missed shot point can only map to a reshoot code, an overlap only to a not-counted code. **A shot with no finding** chooses the production code a clean shot gets.

2.4 Which sequence keeps an overlap

When two sequences of one source line both have a production shot at the same shot point, one is kept and the other is **Overlap**. Keep the **earlier** sequence (the usual choice: the continuation's first shots are the

overlap) or the **later** one, as processing wants it. A shot point the earlier sequence shot badly is not an overlap at all: the later sequence produces it.

3. The routine, line by line

Every change is approved by a named QC. Each dialog that saves a change - a status, an exclusion, a removal or a restore, a sign-off, the scheme - asks for the **QC name**, and the change is signed with it. The last name is kept on this computer, in **QC name** at the top of the window, and offered next time. At Stop, the app works out the findings of the lines just monitored, on a thread of its own; lines recorded before are worked out when the window opens.

3.1 Reviewing a line

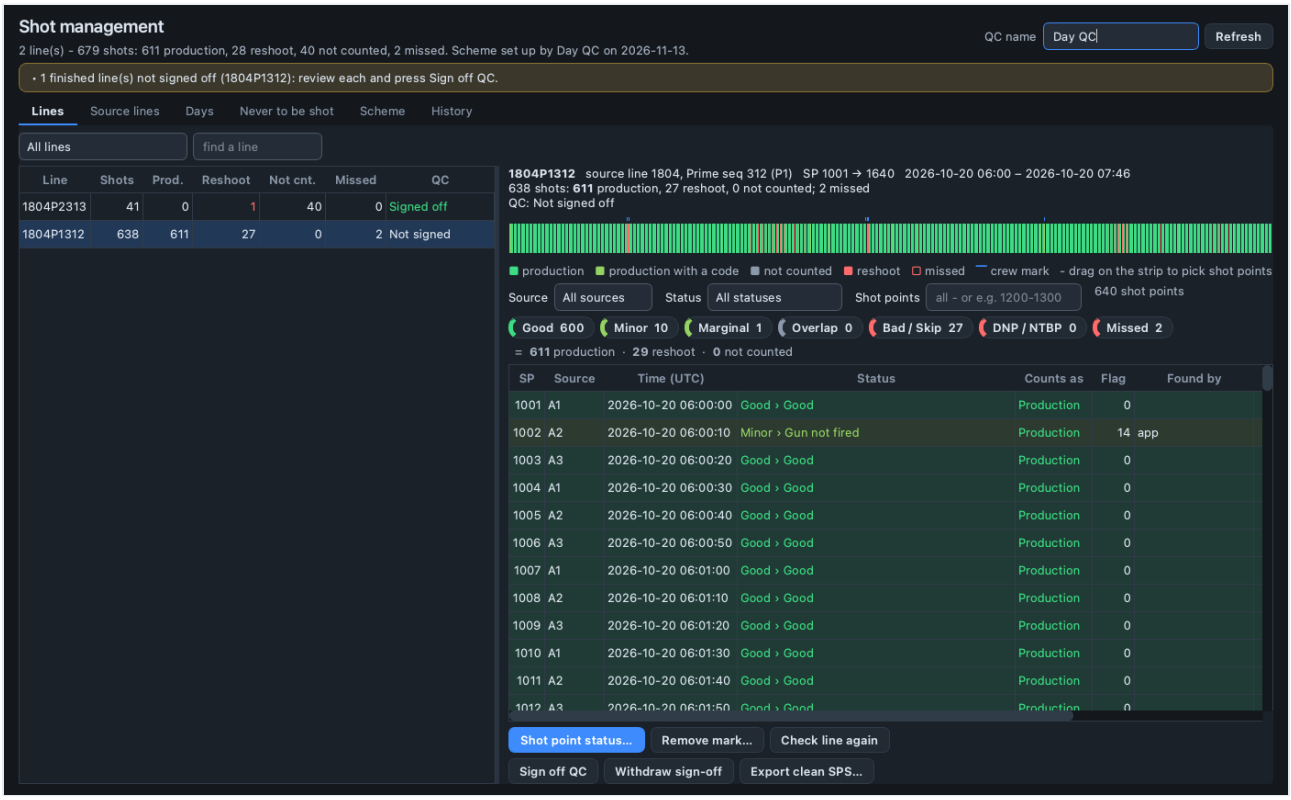

Figure 3. *The Lines page: every line with its production, reshoot, not-counted and missed shots and its QC state; the line picked as a strip of its shot points in the order shot; the filters and a count per class; and every shot point, a row each, tinted by its status. Synthetic data: three sources firing in turn; the second line is a continuation after a circle, so its shots are overlap.*

The **strip** is the line in the order shot, a cell a shot point: green production, a paler green production with a code (a warning, a concession), grey not counted, red reshoot, a red outline a missed point, and a blue tick above a crew mark. Drag along it to pick shot points.

The **table** lists **every shot point** of the line, good ones included, a row each in the order shot, tinted by its status in the strip's colours: the shot point, its **source** (A1, A2, A3 - as the clean SPS names it, from the Source Configuration), the time, the **status** - the code that decides first, any other code the point carries after it (*Overlap* > *Overlap + Gun not fired*) - what it counts as, its flag, who found it (*app*, or the QC who marked it) and why. Right-click a row to select every row with its code.

Above the table:

- **Source** - all sources, or one: A1, A2, A3;
- **Status** - all, production, not production, a class (Good, Minor, Marginal, Overlap, Bad / Skip, DNP / NTBP), missed, or the points with a crew mark;
- **Shot points** - all, or a list or ranges (`1200–1300`);

- a **button per class** with its count - Good, Minor, ... Missed - and the production, reshoot and not-counted totals, over the source and shot points chosen. They follow every change; a click on a class shows that class only, a second click shows all again.

The lines' QC column reads *Not signed*, *Signed off* or *Changed* (since the sign-off); the full words are its tooltip.

The filter above the lines shows all lines, those not signed off, those changed since their sign-off, those with reshoot points, or those with crew marks. The notice under the title says what needs doing: lines finished and not signed off, lines changed since their sign-off, points agreed never to be shot that were shot, lines judged from summaries.

3.2 Changing a shot point's status

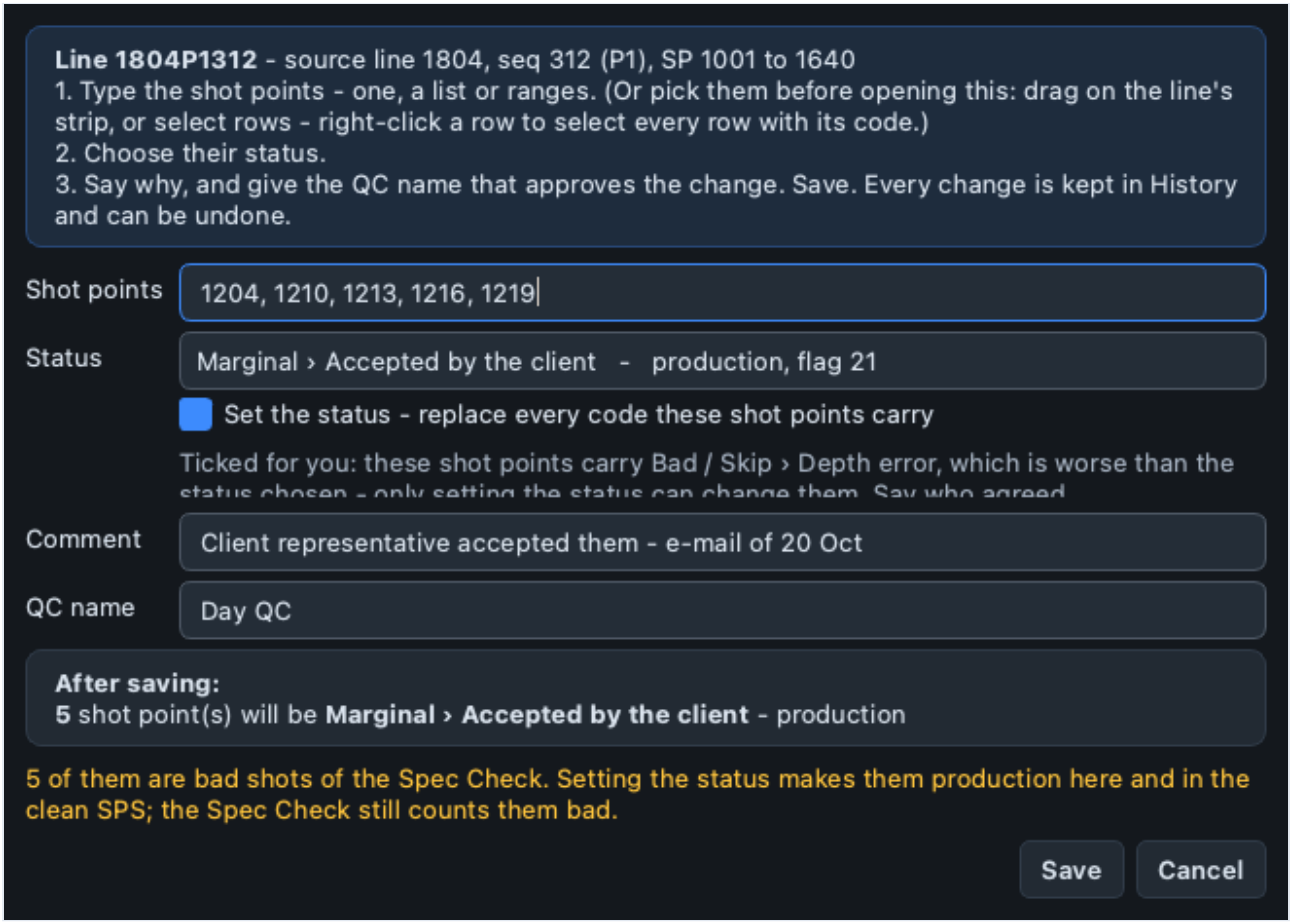

Figure 4. *The client accepts five shots with depth errors. The dialog has ticked Set the status by itself - only replacing what the points carry can make them better - says why, and shows what Save will do.*

To change the status of shot points

1. Pick the line. Pick the points: drag along the strip, or select rows of the table (right-click a row to select every row with its code).
2. **Shot point status…**: the points are filled in - edit them if needed, or type them: one, a list or ranges, `1290, 1294–1296` (also `1294 to 1296`; a reversed range is turned round).
3. Choose the status, write why, and give the **QC name** that approves the change. Read *After saving*: what each point will be, and any point that stays as it is and why.
4. **Save** is possible once *Still needed* is empty and nothing is shown in red.

The two usual cases:

- **Mark extra bad shots** - choose a bad status (Bad / Skip > Noise, say) for any points, clean ones included. It is **added** to what the points carry and the worst decides, so marking a bad shot "Noise" does not hide its timing error - both are kept, and the worse is its final code.
- **Accept shots the app found bad** - choose a better status (Marginal > Accepted by the client). Added, it would change nothing, because the worse code decides; so the dialog ticks **Set the status** for you, which replaces what the points carry, and asks who agreed (§4.6). Untick it if you only wanted to add the code.

Marks apply in the order they were made: setting the status replaces the app's codes and every earlier mark on those points, and a mark made after it adds again - a bad shot found after a concession still counts. **Remove mark…** takes the crew marks of the selected rows off; nothing is deleted (§4.11). The app's own findings cannot be removed - setting the status answers them.

3.3 Signing off and the clean SPS

Sign off QC asks for the QC name that approves it and records that the line's QC is finished, with that name and the time, and a fingerprint of how every shot point of the line is classified. If anything changes afterwards - a mark, the scheme, the line re-run, a point excluded - the line reads **Changed since sign-off** in the list and the notice: look at it again and sign it off again. **Withdraw sign-off** takes it back. A line being recorded cannot be signed off.

Export clean SPS… writes the line's **production shots only**, as SPS S records, into the project's `clean_sps` folder, with the line's full shot list as CSV beside it. A line not signed off, or changed since, is written as a **DRAFT**: the files' names end in `_DRAFT` and the header says so. Each record carries:

- the **source line** and the **shot point**, a point index (1, or 2 where two sources shot one point) and the **point code** set on the Scheme page - `A{n}` (the default) writes each shot's source, A1, A2, A3; a fixed code such as `A1` is written on every shot;
- the **preplot's coordinates** - planned positions, not navigation; blank where the project has no preplot for the line;
- the **day of the year and time** of the SEG-D header, in UTC.

It is written in the preplot's SPS layout (the layout SPS had when the SEG adopted it, or revision 2.1); without a preplot, in the first. H26 records say what the file is, how many production shots of how many recorded, who signed the QC off and when.

4. Situations, and what to do

4.1 A line interrupted and continued after a circle

The vessel stops a line, circles and starts again a few shot points before where it stopped, under a new sequence of the same source line (1804P1312 continued as 1804P2313 - both source line 1804, by the digits before the letter). Nothing to do: where both sequences produced a point, the app keeps one (§2.4) and the other's shots are **Overlap** - not counted, not in its clean SPS, not a hole. The Source lines page lists both attempts.

4.2 A marine mammal shutdown, or a stop for another reason

The points not fired during a shutdown are **missed** - no record. Mark them with the reason's code (Marine mammal shutdown, Obstruction, …) and a comment: they stay to be shot, and the reason travels with them into the report, the CSV and the handover.

4.3 Shots rejected, DNP or NTBP, to be shot again

Mark them with a reshoot code (Bad / Skip, Do not process). They are not in the clean SPS and not in the day's production, but the time they took is shooting time. A whole line marked **not to be processed** (in the End of Line report's comments box) has every shot *Not to be processed*: its clean SPS is empty and its points are left to shoot.

4.4 Reshoots and infills

An R or I line of the same source line shoots the holes again. Its production shots produce those points; where the earlier sequence already produced a point, the reshoot's shot is an overlap (or the earlier one is, with *keep the later*). The Source lines page shows what is still left.

4.5 Run-in and run-out shots

With a preplot, a recorded shot point the preplot does not plan - before the line's first point or after its last - is **Run-in / run-out**: not counted, not in the clean SPS. Without a preplot the app cannot tell; mark them by hand if the contract wants them out.

4.6 The client accepts shots outside a limit

Say the client agrees to keep a line's shots with depth errors. Right-click a Depth error row and choose *Select every row of «Bad / Skip > Depth error»*, press **Shot point status…** and choose Marginal > Accepted by the client. The dialog ticks **Set the status** for you: the app's codes for those points are replaced, a comment is required (who agreed, and why), and the app's own finding stays written beside it. The Spec Check still counts the shots as they were - setting the status changes production and the clean SPS, never the app's QC.

4.7 Part of a line agreed never to be shot

On **Never to be shot**, **Add…**: the source line, the points, a code, a description of the area and why, who approved it for the client, the reference and the date. The points leave the plan everywhere: not missed shots (End of Line report, SP Quality, the live counter), not holes in the Survey, not left to shoot. With a preplot, the source line and the points must be in it. A point excluded that was shot as production stays counted, and the notice says so - check the exclusion.

4.8 A line recorded again (re-run)

A re-run replaces the line's data, and its findings are worked out again. The crew's marks stay; its sign-off reads *Changed since sign-off* if anything moved.

4.9 A line's shot data deleted, or a test line

Deleting a line's shot data (Manage Lines) keeps its findings, its marks and its per-shot summaries: its classification stays. A line whose findings were never worked out before its data went is judged from its summaries - no nav mask or firing sync check - and the window says so. A test line taken out in Gun Health (Lines tab) leaves shot management too.

4.10 Two people, two computers

Each QC approves with their own name. Marks are kept apart however they are made; press **Refresh** to see what someone else did.

4.11 A mistake

A wrong mark: select its row, **Remove mark…**, and mark again. Nothing is deleted - **History** lists every mark and exclusion ever made, with who and when, removed ones included; select one and **Restore**. The status dialog refuses shot points that are not on the line and a range longer than the line itself (a typing slip such as 1290-12960); what is merely not filled in yet - the points, the status, a comment - it lists quietly under *Still needed*.

4.12 Changing the scheme during the survey

Rename and describe freely: a mark holds its code, not its name, so a rename renames every row. Changing what a class counts as changes every shot of its codes, so the page says how many marks it touches; the lines signed off then read *Changed since sign-off*. A code in use cannot be deleted - hide it, or move its marks first.

5. Where the figures show

5.1 The line by line history

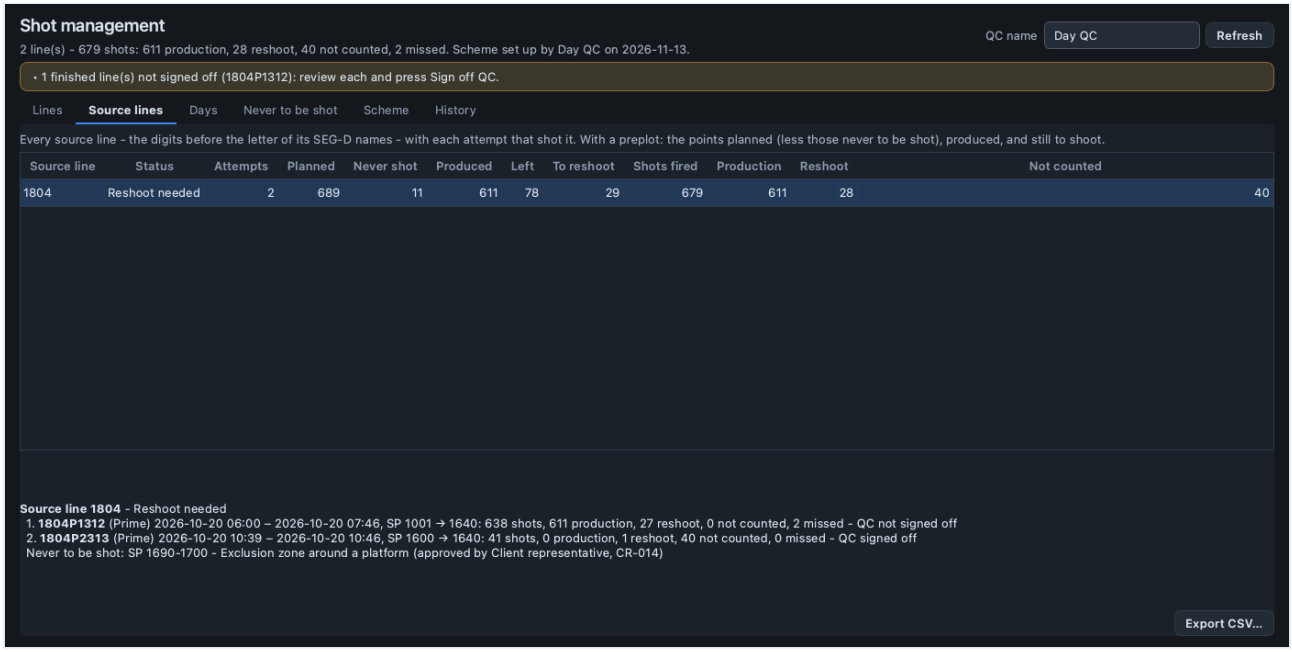

Figure 5. *Source lines: each source line with its attempts, and - with a preplot - the points planned (less those never to be shot), produced, left and to be shot again. Below, the chosen line's attempts one by one, with their QC state, and its exclusions.*

For every source line: its **attempts** (the SEG-D lines that shot it, in the order shot, with their times, shot points, counts and QC state), the shots fired, production, reshoot and not counted; and with a preplot the points **planned** (the preplot's, less those never to be shot), **produced**, **left** and **to be shot again**, and its state: complete, reshoot needed, partly shot, to shoot, never to be shot. A preplot line not shot yet is listed *To shoot*. **Export CSV…** writes the table.

5.2 Days

A row a day (00:00 to 24:00 on the handover's SEG-D clock): the lines, the shots fired, the **production** shots, those to reshoot and those not counted, the production share, the time shooting, in line changes and stopped. A day counts its production shots only - reshoot and overlap shots are not production - but the time they took is shooting time. **Export CSV…**.

5.3 The Survey, the report and the handover

- **Survey** - once shot management is set up, every figure is production: a line's and a day's clean shots are its production shots (the card reads *Production shots*), only a production shot covers a preplot point - a point shot only with rejected shots stays a hole - and points never to be shot leave the plan.
- **End of Line report** - a *Shot Management* page for each line: its shots by what they count as, by code, its QC sign-off and the points never to be shot on its source line; then every run of shot points that is not plain production, with who marked it and why.
- **Shift handover** - each of the shift's lines as production, reshoot, not counted and missed with its QC state, and every mark and exclusion made or removed in the shift.

6. What the app stops you doing

YOU TRY TO	THE APP
Save a scheme with two classes, or two codes of one class, of one name (whatever the case), or two codes with one short word	Refuses, and names them
Give two classes one flag, or a code a flag outside its class's band	Refuses, and gives the band
Put a production class above a not-counted or reshoot class	Refuses
Map a Spec Check bad shot to a production code	Refuses unless you tick that you allow it
Map a missed point to anything but a reshoot code, or an overlap to anything but a not-counted code	Refuses
Delete a code or class that a mark, a finding or the clean shot uses	Refuses; offers Hide and Move marks
Save any change without a QC name	Asks for it; nothing is done without one
Mark points that are not on the line, or a range longer than the line	Refuses, and gives the line's shot points
Set the status without a comment, or use a code that asks for one without	Lists it under Still needed; Save waits
Choose a better status than the points carry, without setting it	Ticks Set the status for you, and says why
Give a missed point a production code	Warns; the point stays missed
Set a Spec Check bad shot to a production status	Warns that the Spec Check still counts it bad
Exclude points without an approval, or points not in the preplot	Refuses
Exclude points already shot as production	Warns; they stay counted
Sign off the line being recorded	Refuses until Stop
Export the clean SPS of a line not signed off, or changed since	Writes a DRAFT, said in the name and the header
Lose a mark by removing it	Keeps it in History, to restore

7. Reference

7.1 The standard scheme

CLASS (FLAG)	COUNTS AS	CODES (FLAG, SHORT WORD)
Good (0)	Production	Good (0, OK)
Minor (10)	Production	Timing warning (11, TW), Depth warning (12, DW), Pressure warning (13, PW), Gun not fired (14, NF), Autofire (15, AF)
Marginal (20)	Production	Accepted by the client (21, CON), Marginal - other (29, MO)
Overlap (30)	Not counted	Overlap (31, OV), Run-in / run-out (32, RI)
Bad / Skip (50)	Reshoot	Timing error (51, TE), Depth error (52, DE), Pressure error (53, PE), Source dead (54, SD), Nav mask error (55, NM), Time-break sync (56, TB), Firing spread (57, FS), Missed - no record (58, MS), Misfire (60, MF), Seismic interference (61, SI), Noise (62, NO), Navigation (63, NV), Recording system (64, RS), Marine mammal shutdown (65, MM), Other (69, OT)
DNP / NTBP (70)	Reshoot	Do not process (71, DNP), Not to be processed - line (72, NTBP)
Never to be shot (90)	Never to be shot	Client approval - obstruction (91, COB), Client approval - exclusion zone (92, CEZ), Client approval - other (99, COT)

These are a starting point, not a standard: set the contract's own.

7.2 The app's findings

FINDING	WHERE IT COMES FROM
Fire time, depth, pressure outside its limit	The Spec Check's rule: at least its minimum number of guns of the shot outside the limits in their own header
Source dead, nav mask, time-break sync, firing spread	The Spec Check's rule, each where its criterion is on
Missed - no SEG-D record	A point on the line's grid with no record, planned (not excluded, and in the preplot where the preplot can judge)
Warnings	A reading outside its limit on fewer guns than make a bad shot; a gun that did not fire while its source fired; an autofire
Overlap	Another sequence of the same source line produced the point
Run-in / run-out	A recorded point the preplot does not plan
Not to be processed	The line is marked NTBP

7.3 The files

FILE	WHAT IT HOLDS
`clean_sps/<line>_clean.sps`	The line's production shots as SPS S records (`_DRAFT` when the line is not signed off, or changed since)
`clean_sps/<line>_shots.csv`	Every shot point of the line, recorded or missed: what it counts as, its final code, class and flag, the app's and the crew's codes, override, comments
Source lines, Days	**Export CSV…** of each page

> The project database keeps the scheme, every mark and exclusion (removed ones too), each line's findings and the sign-offs. Marks and exclusions are kept when a line is re-run or its shot data deleted; another project's scheme can be copied, never its marks.