Asset Data Checking and Population Guide

How asset data gets from your Revit models into Deep Space, how it is checked in LOI Advanced, and how it becomes the asset register in the Asset Portal.

This guide explains the whole asset data journey in Deep Space as one process: the Revit sync, the automatic post-sync pipeline that creates assets, the data checks in **LOI Advanced**, and the asset register and child assets in the **Asset Portal**. It is written for digital engineering coordinators, BIM managers and the trades who populate asset data. The screenshots are from a live project (October 2026).

How the pieces fit together

Step	Where it happens	What it does	Who does it
1. Model the asset data	Revit (the live models on ACC)	Modellers and trades fill in the asset parameters (for example `ATC_EquipmentName`, `ATC_Category`, `ATC_Floor`) on each element	Modellers, trades
2. Sync	Deep Space Revit add-in (scheduled)	Reads the models and uploads elements and their parameter values to Deep Space	Automatic (scheduled), Deep Space
3. Post-sync pipeline	Deep Space cloud	For each model that finishes uploading, decides which elements are assets, copies their asset data, and updates the register	Automatic
4. Check	DS Pro ▸ **LOI Advanced**	Shows, element by element, which required parameters are missing, blank, invalid or valid	Digital engineering coordinator
5. Use the register	DS Pro ▸ **Asset Portal**	The asset register: every asset with its data, plus non-modelled **child assets** added by the trades, and CSV export	Coordinator, trades

The loop is simple: **fix data in Revit → sync → check in LOI Advanced → repeat**, while the Asset Portal always shows the register built from the latest sync.

> **One rule to remember:** Deep Space only ever shows what was in the model at the last successful sync. If a value was changed in Revit after that, it appears after the next sync.

Part 1: The Revit sync

What is synced

The Deep Space Revit add-in opens each model and uploads, for every model element:

- the element itself: its category, family, type, level and workset, plus its unique id and element id;
- its **parameter values** (instance and type parameters);
- rooms, views, sheets and model warnings.

For a federated model, the add-in opens the host model and then each linked model in turn. Every model that finishes uploading is stamped **upload complete**, and that stamp is what starts the post-sync pipeline (Part 2).

How the sync runs

Syncs normally run **unattended on a schedule**, on a dedicated machine managed by Deep Space, against the live published models on ACC. You do not need to send models to Deep Space; we always read the latest published version.

Two sync options matter for asset data:

Option	What it does	When to use it
Only sync changed files	Skips any model that has not been published since its last sync	On by default. It saves hours on large federations, because unchanged models are not re-read
Only sync whitelisted parameters *(new, October 2026)*	Reads and uploads only the parameters on the project's **parameter whitelist** instead of every parameter on every element	Large models whose full sync takes too long. Used for Built's federated syncs since 13 October 2026

The parameter whitelist was introduced in October 2026 because the largest services models carried up to 26 million parameter values, most of them never used. With the whitelist, the hydraulics model's parameter read fell from about 18.6 million values to 7.1 million, and its sync completed for the first time since May. The fire model read its parameters in 33 minutes; a week earlier its whole sync had taken 14.6 hours.

What is on the whitelist. The whitelist is built from the parameters Deep Space actually reads:

1. every parameter mapped to an LOI attribute (the parameters LOI Advanced checks and the Asset Portal displays);
2. the location parameters used for levels (`Level`, `Reference Level`, `Base Level`, `Base Constraint`) and `IfcGUID`;
3. a small set of identity parameters for model checking (`Category`, `Family`, `Type`, `Workset`, `Mark`, `Comments` and similar).

> **Adding a new asset parameter?** If you add a parameter to the LOI requirements, ask Deep Space to refresh the whitelist and re-sync. Until then, the new parameter will show as *Does Not Exist* in LOI Advanced, even if it is filled in in Revit.

How to tell a model has synced

- **LOI Advanced ▸ Model Register** lists every model with its **Last Synced** date (see Part 3).
- **Asset Portal ▸ file filter** lists every model that has at least one asset. A model that has never synced since asset escalation was switched on is not in this list.

> **Note on "Not Synced" for unchanged models.** When *Only sync changed files* skips a model because nothing changed, the model keeps its previous data, which is still current. Its **Last Synced** date is updated, but the Model Register may still show it as **Not Synced** because no new data was read within the recency window. That is expected.

Part 2: The post-sync pipeline (how assets are created)

Every time a model finishes uploading, Deep Space runs an automatic job for **that one model**. The job refreshes the project's **Project Library** (every Revit family and type in the project, its elements, and their asset data), decides which elements are **assets**, and updates the Asset Portal. Nobody needs to start it.

```
Model finishes uploading ("upload complete")
   -> one job for that model
   -> new-data check  --nothing new-->  stop
   -> refresh families, types and elements; read room, level and category
   -> copy the asset (LOI) parameter values
   -> remove assets whose elements have left the model
   -> create new assets (two routes, below)
   -> give new assets their attribute set and register fields
   -> Asset Portal updated
```

When the job runs, and when it does not

Situation	What happens
A model syncs with changes	The job runs all its steps for that model, usually within a few minutes of the upload finishing
The model was skipped as unchanged (*Only sync changed files*)	The job finds no new data and stops. Nothing changes in the Asset Portal for that model, which is correct, because nothing changed in the model. One exception: a rule switched on since the model's last real sync (for example elemental escalation) is not applied until the model is next published, or until Deep Space runs the job manually
Deep Space runs the job manually	Every step runs on the model's latest data, whatever its age. This is how existing models are brought up to date after a rule change
One step fails	The other steps still run. The failed step is retried at the model's next sync

Almost every step works on **the model that synced** only. Assets in other models are not touched.

What makes an element an asset

An element becomes an asset by either of two routes. Both are checked on every sync.

Route 1: the element has an Equipment Name (elemental escalation). Switched on for Built on 2 October 2026 with the Asset Portal setting *Escalate all elements that have this attribute populated* (Part 4). An element becomes an asset when **all** of these are true:

1. the element is in the model that just synced, and that model is active in the project;

2. the element's family/type has an `ATC_Category` value that **exactly matches an attribute set**: *Architecture, Control, Electrical, Fire, Hydraulic, Mechanical* or *Vertical Transportation*;
3. its `ATC_EquipmentName` is filled in;
4. the value is not a placeholder. These are ignored, regardless of capitals or surrounding spaces: blank, `-`, `--`, `---`, `___`, `n/a`, `na`, `none`, `tbc`, `tba`, `false`, `0`, `.`;
5. it is not already an asset.

Route 1 uses the value **in the model**. Typing an Equipment Name in the Asset Portal does not make an element an asset. It does not change anything about the element's family or type either, so un-named elements of the same type stay out.

Route 2: the element's family and type are marked as assets. In the Project Library, a Revit family/type can be marked **is asset**: automatically, when it exactly matches one of the project's family/type rules (Asset Portal ▸ Settings ▸ Auto Creation, 119 rules on Built), or by hand. Marking a type creates an asset for every element of that type straight away.

> **Practical rule:** the dependable way to make sure an element is in the register is to give it a real `ATC_EquipmentName` and the right `ATC_Category` in Revit.

What never becomes an asset

- annotation and other non-model elements (views, sheets, tags);
- elements in models that are no longer active in the project;
- elements whose `ATC_Category` does not match an attribute set (for example `-`, `TFNSW-MODEL-OBJECTS`, or a combined value such as `Mechanical, Filtration Unit`), unless their type is marked as an asset;
- a second asset for the same element in the same model.

Where each field in the register comes from

Field	Comes from	Updated on each sync?
Equipment Name	`ATC_EquipmentName` in the model, unless someone has typed a different value in the Asset Portal	Yes. A value typed in the Asset Portal always wins, and it is never written back to Revit
Classifying parameter (category)	`ATC_Category` on the element's family/type	Yes. It is held once per family/type, so if elements of one type carry different values, the last one read wins
Revit category, family, type, element ID	The model	Yes
Attribute set (which fields the asset has)	The attribute set whose name matches `ATC_Category`	**Set once**, when the asset is created. Changing `ATC_Category` later does not move an existing asset to another set. Ask Deep Space if one needs moving

Field	Comes from	Updated on each sync?
Asset data (the LOI fields)	The matching parameters in the model	Yes, including blanks: a value cleared in Revit is cleared in Deep Space. Only for types whose `ATC_Category` matches a set
Level and **Room** (register fields)	The element's level and room **when the asset was created**	No. They are a snapshot. The asset's Relationships tab shows the element's current level
Facility description, building, installation contractor	The Revit family name, the model's building code, the project default	No. Set when the asset is created

When assets are removed

What happened in Revit	What happens to the asset
The element was deleted	The asset is removed at that model's next sync. A backup is kept, so it can be recovered
The element was changed to a different family or type	Deep Space treats it as a new element. The old asset is removed, and a new one is created if the element still qualifies. **The new asset has a new Asset ID, and values typed in the Asset Portal against the old asset do not carry over**
A family/type was un-marked as an asset	Every asset of that type is archived and removed. If the type matches an Auto Creation rule, it is marked again at the next sync
`ATC_EquipmentName` was later cleared, or set to a placeholder	Nothing. The asset stays
The model was skipped as unchanged, or made inactive	Nothing

> **Keep Asset IDs stable:** avoid swapping an element's family or type once trades have started adding data or children against it. Edit the existing type's parameters instead.

Things to know

- `ATC_Category` **must exactly match an attribute set name.** Any other value means no asset data is read for that type, no elemental escalation, and no attribute set (shown as *-1*). LOI Advanced shows these elements under the condition *(Blank)*. Near misses are the most common cause: in October 2026 LOI Advanced showed elements classified **Controls** (the set is *Control*), **Vertical Transport** (the set is *Vertical Transportation*) and **Mechanical, Filtration Unit**.
- **Room and level depend on the room-detection data from the model sync.** Elements added or moved since rooms were last detected may have no room. Each family's Room Calculation Point must sit inside its room.
- **Changes appear after the next sync of that model**, not when another model in the federation syncs.

- **The latest sync always wins.** If a parameter is missing from a sync altogether (rather than blank), the previous value is kept.

Part 3: Checking asset data in LOI Advanced

LOI Advanced checks every element that should carry asset data against the project's **LOI requirements**: which parameters each kind of element must have, and what values are acceptable.

Opening LOI Advanced and refreshing it

Open Horizon, select the project, then **DS Pro ▸ LOI Advanced**. The app has seven tabs across the top and three tools at the top right: **Full screen**, **Generate PDF** and **Refresh Dataset**.

> **Refresh after a sync.** LOI Advanced is a report: it shows the data as of its **last refresh**, not live. After a sync has finished, select **Refresh Dataset** (the circular-arrows button). The refresh takes a few minutes. Reload the page when it is done. The refresh time is shown at the bottom of the Overview tab.

Every tab shares the same filters: **File Name Search**, **Element ID Search**, **Package**, and the **Filter** panel (Discipline Group, File Discipline, Parameter, Workset / Layer, RvtFileName, File Name, Category, Level, Room, **Classifying Parameter**). Filters apply across the tabs, so you can narrow to one model or one package and move between tabs.

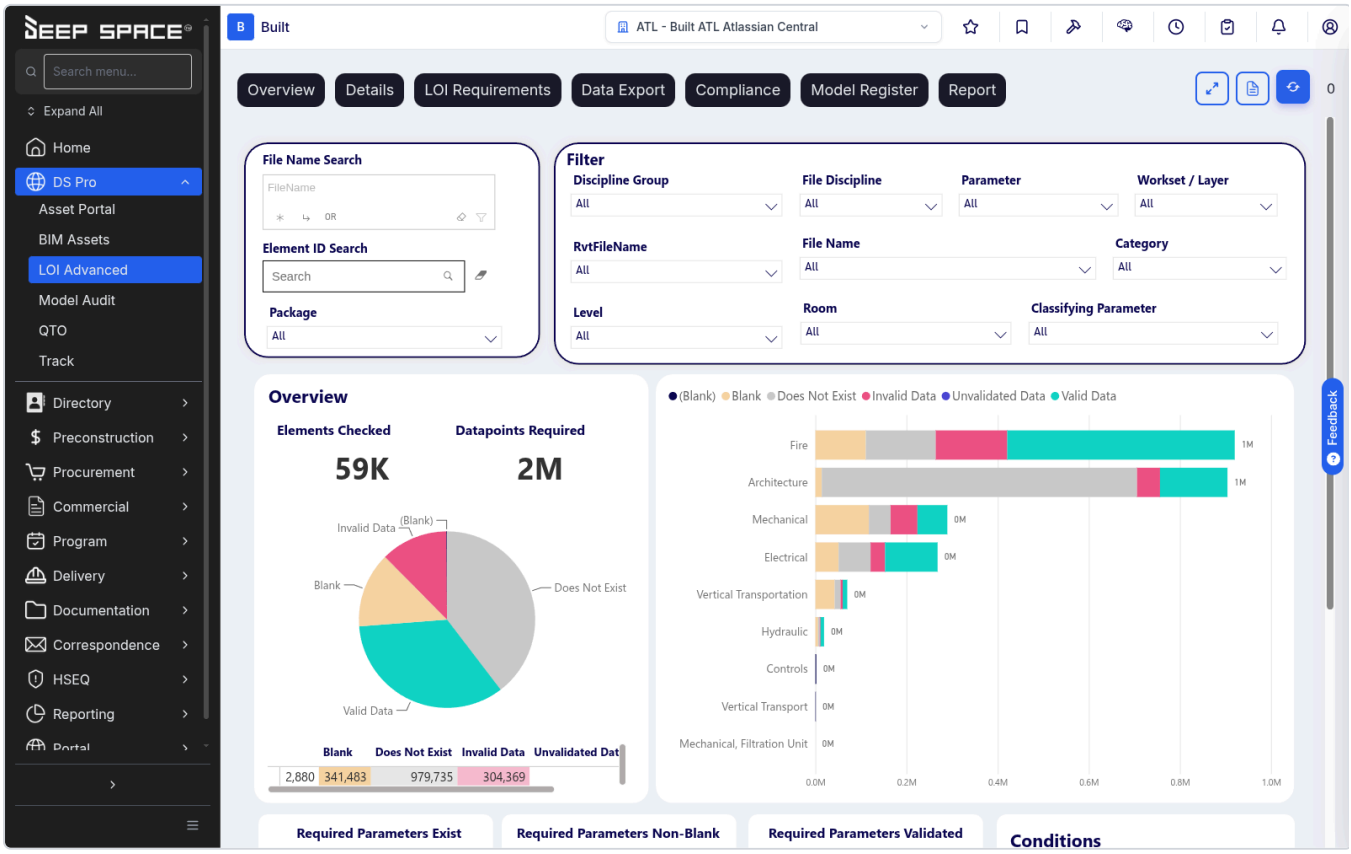
The five conditions

Each required parameter on each element gets exactly one condition:

Condition	What it means	How to fix it
Does Not Exist	The element needs this parameter, but the parameter is not on the element at all	Add the shared parameter to the family or project in Revit, then fill it in
Blank	The parameter is on the element but empty (or just "-")	Fill in the value in Revit
Invalid Data	The parameter has a value, but it fails the validation rule (for example, not in the allowed list, or the wrong format)	Correct the value to one that passes. The **LOI Requirements** tab shows the rule
Valid Data	The parameter has a value that passes its validation, or needs no validation	Nothing to do
(Blank)	The element's classification does not match an attribute set, so no requirement could be applied	Correct the classifying parameter value (for Built, `ATC_Category`) to an exact attribute set name, for example *Control* rather than *Controls*

Which parameters an element needs depends on its **classifying parameter**. On Built this is `ATC_Category`: its value (Fire, Mechanical, Electrical, Hydraulic, Vertical Transportation, Architecture, Controls …) selects an **attribute set**, which lists the required parameters for that kind of element.

Overview

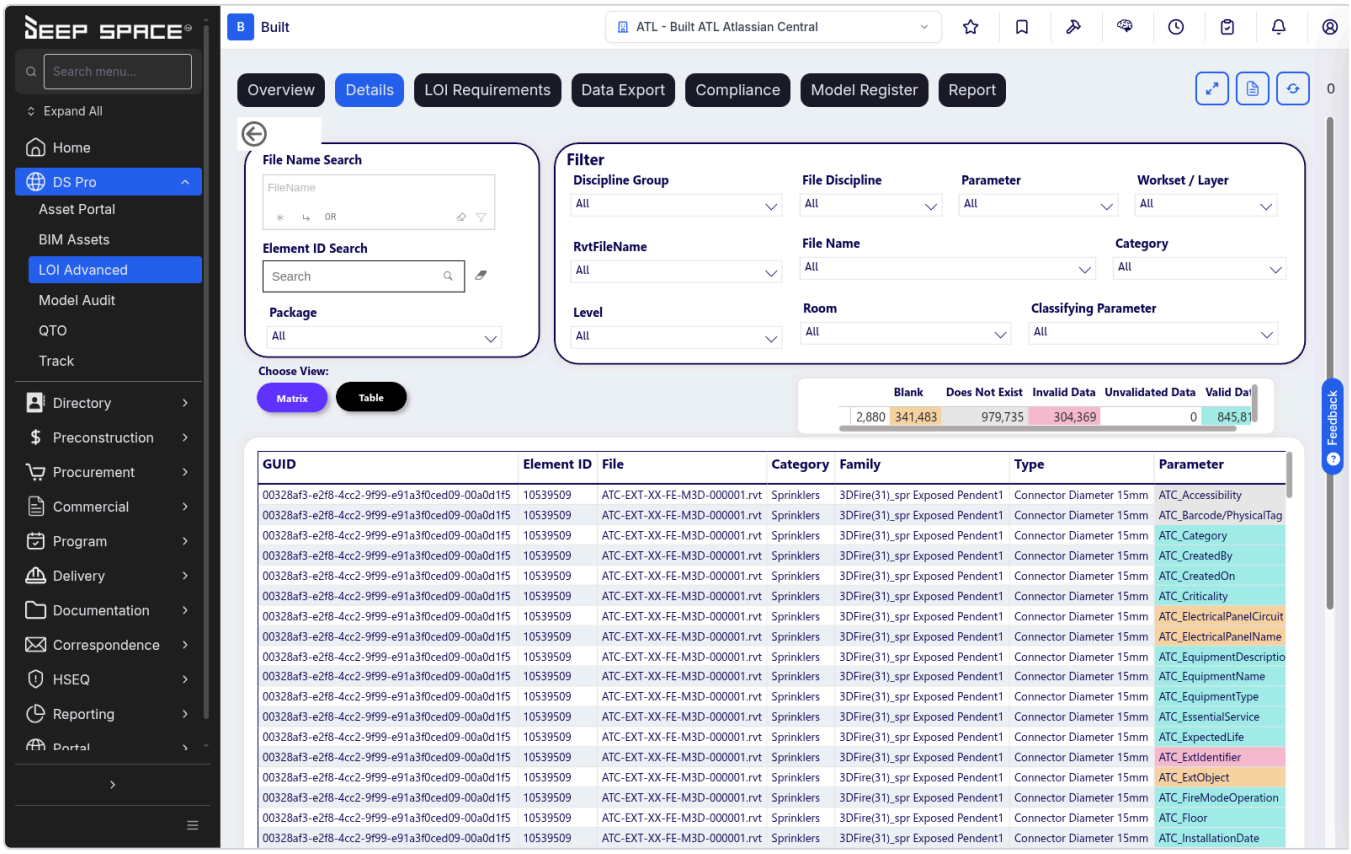


Overview tab: elements checked, data points required, the condition split, results by classification, and three progress gauges.

The Overview answers "how complete is our asset data?":

- **Elements Checked**: how many elements have a classification that requires asset data.
- **Datapoints Required**: elements × required parameters.
- the **pie chart** and the table below it: the split of all data points by condition;
- the **bar chart**: the same split per classification, so you can see which discipline needs the most work;
- three gauges further down: **Required Parameters Exist**, **Non-Blank** and **Validated**, as percentages.

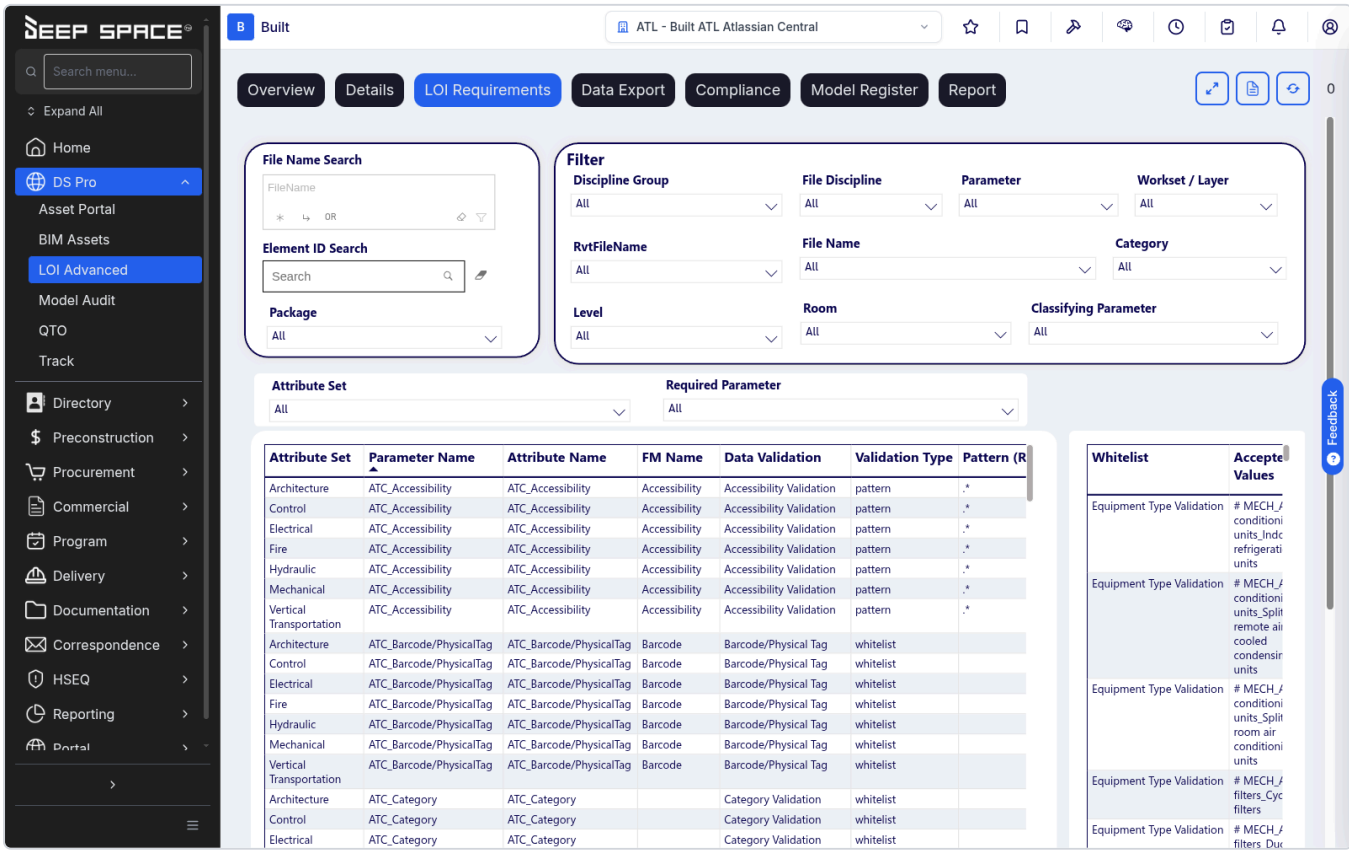
Details



Details tab: one row per element and parameter, coloured by condition.

Details lists every element and required parameter with its GUID, element ID, file, category, family, type, parameter and value, coloured by condition. Use **Element ID Search** to find one element (the element ID matches Revit's *Select by ID*), or **Choose View: Table** for a flat list you can scroll. This is the tab for fixing things one element at a time.

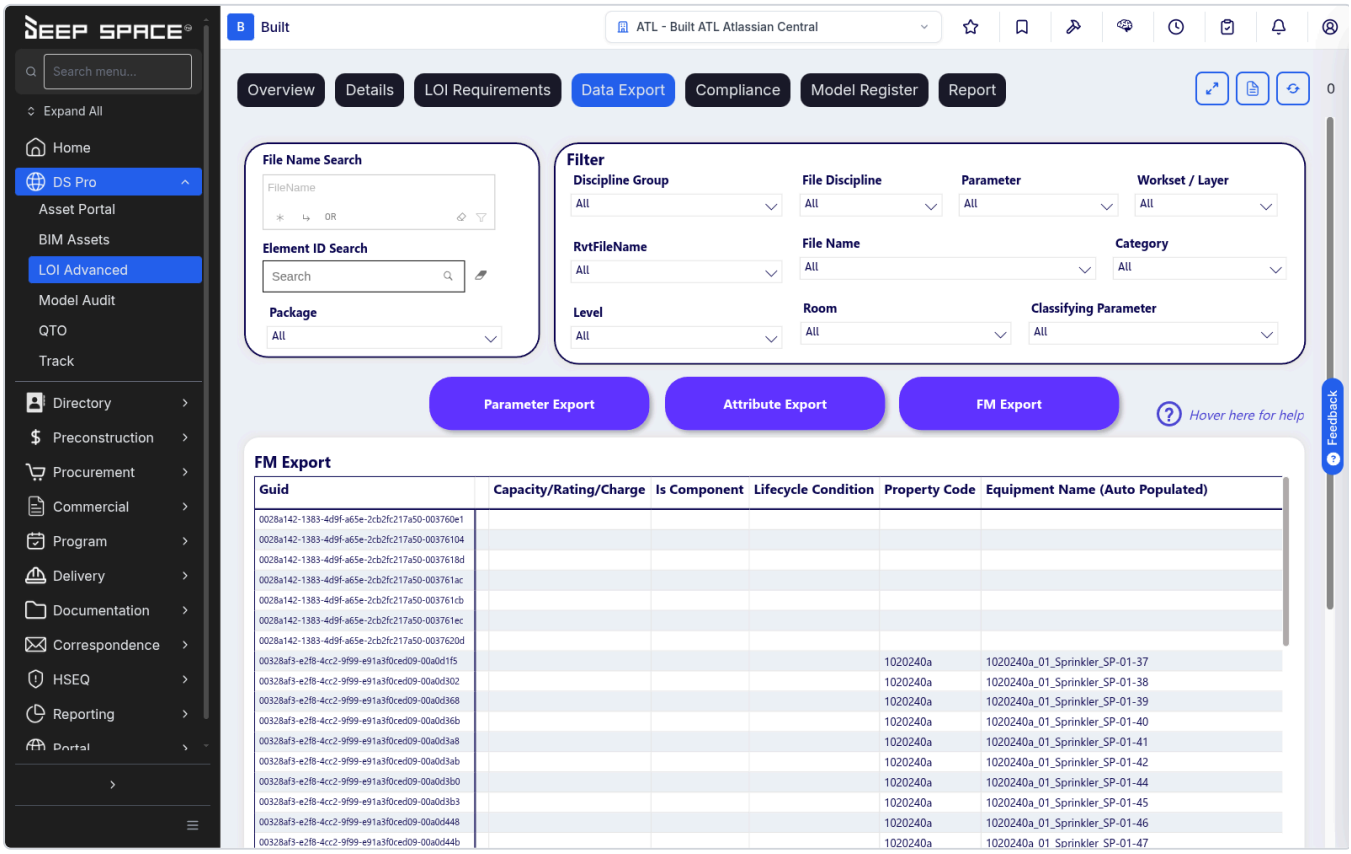
LOI Requirements



LOI Requirements tab: each attribute set's required parameters with their validation rules, and the whitelist of accepted values.

This tab shows the rules: for each **attribute set**, the required **parameter**, its **data validation**, the **validation type** (a *pattern* such as a number or date format, or a *whitelist* of accepted values) and, on the right, the accepted values for each whitelist. When something is *Invalid Data*, look here to see what is accepted.

Data Export

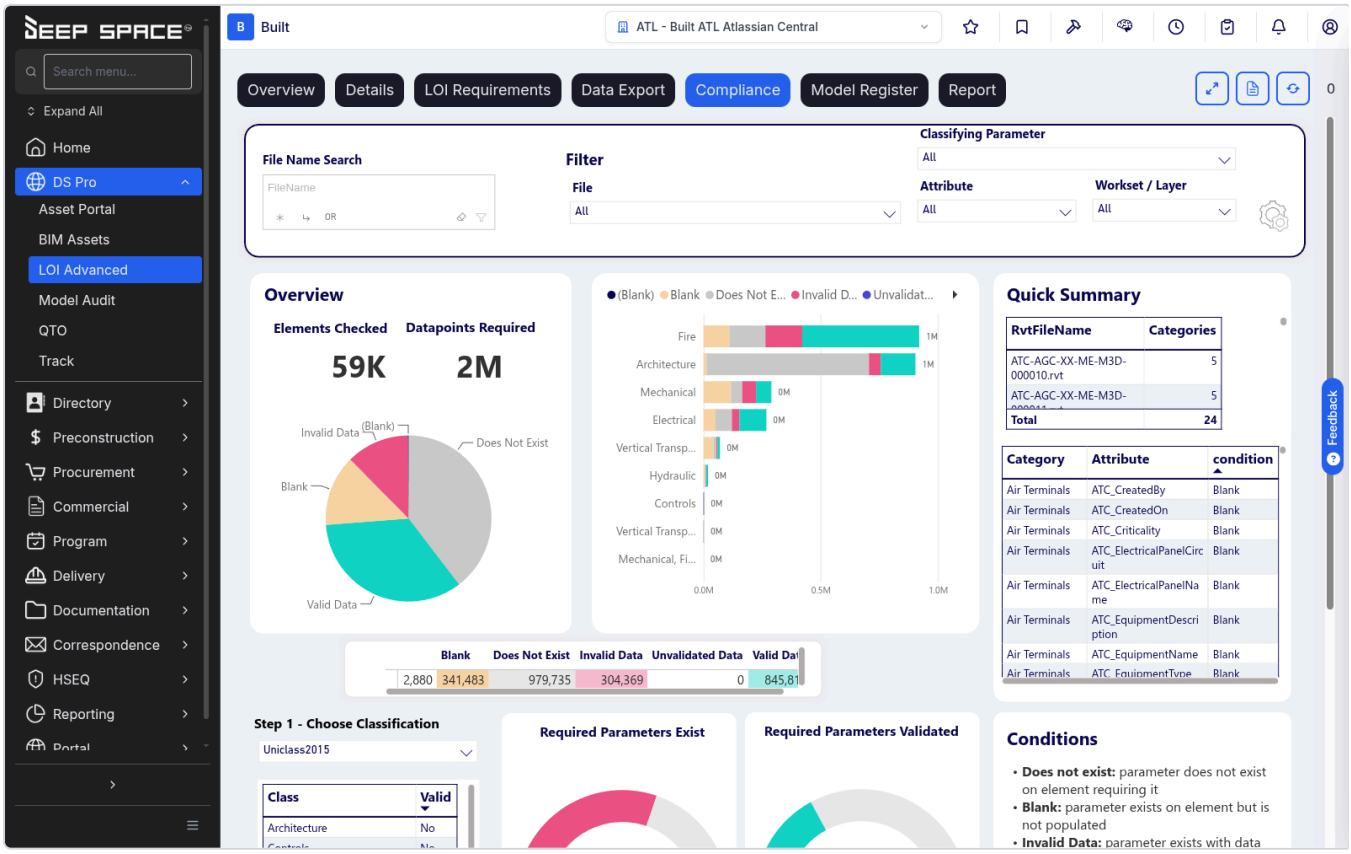


Data Export tab: Parameter Export, Attribute Export and FM Export.

Three exports of the filtered data:

- **Parameter Export**: one row per element and parameter, as checked;
- **Attribute Export**: the same data arranged by LOI attribute;
- **FM Export**: one row per element in the facility-management (asset register) column layout, including the auto-populated Equipment Name.

Compliance

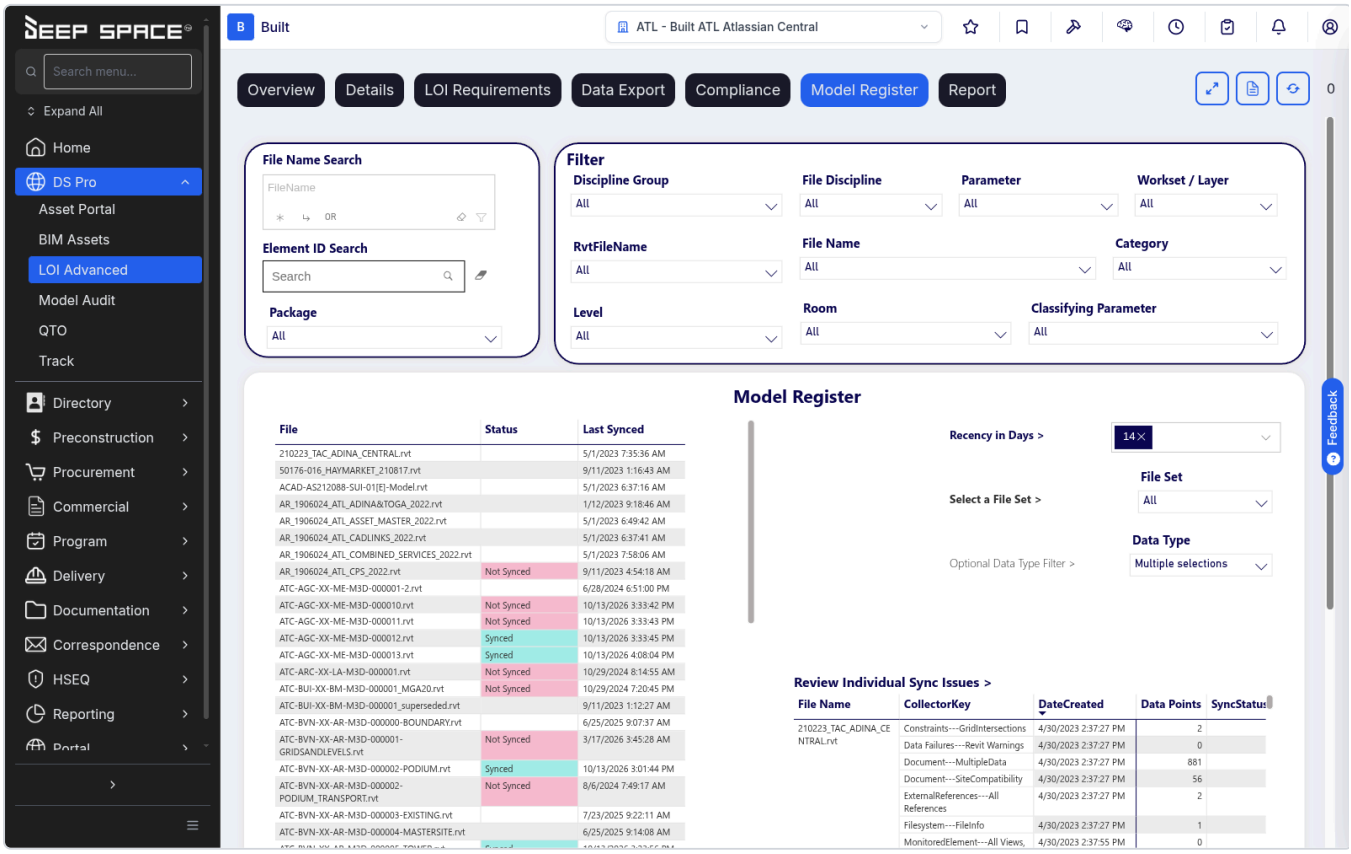


Compliance tab: overview, results by classification, a quick summary by file, and conditions by category and attribute.

Compliance is the management view. It shows the same condition split, a **Quick Summary** per file, the condition of each **category and attribute**, and (lower down) a classification check against a chosen coding scheme such as Uniclass 2015. Use it to agree with each trade which categories and attributes they still owe.

> This **Compliance** tab is about asset data. It is not related to company compliance in the Company Directory, which is covered in *How Contractor Compliance Works in Deep Space*: https://cdn.deepspacesync.com/help-guides/How%20Contractor%20Compliance%20Works%20in%20Deep%20Space.pdf

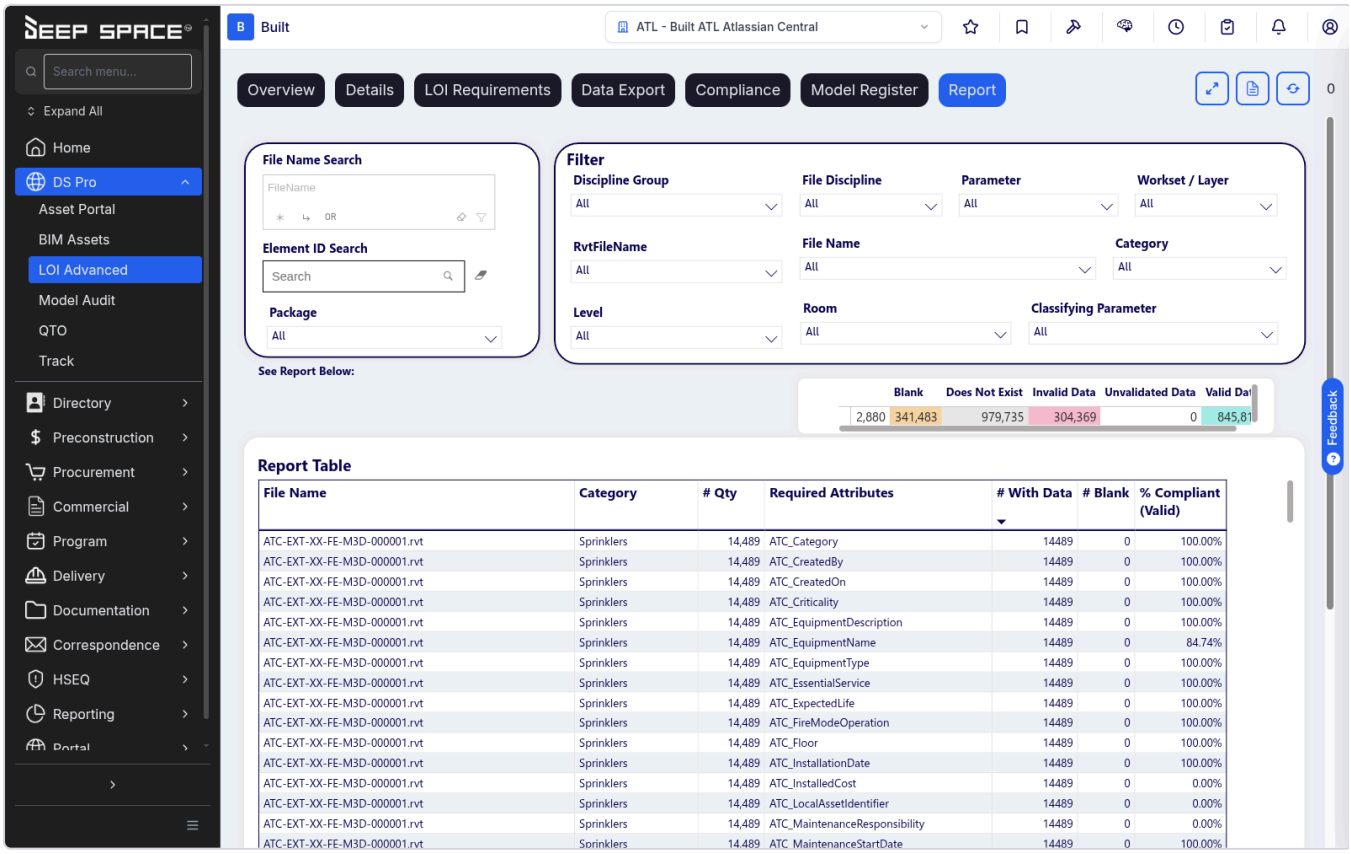
Model Register



Model Register tab: every model, its sync status and last-synced date, with the recency window and sync issues.

The Model Register lists every model in the project with its **Status** and **Last Synced** time. **Synced** means fresh data was read within the **Recency in Days** window (14 days by default); **Not Synced** means it was not. See the note in Part 1 about unchanged models. **Review Individual Sync Issues** on the right lists, for a selected model, each data collector with its date and number of data points. Use it to check that a model has fully uploaded.

Report



Report tab: per file, category and required attribute, the quantity, how many have data, how many are blank, and the percentage compliant.

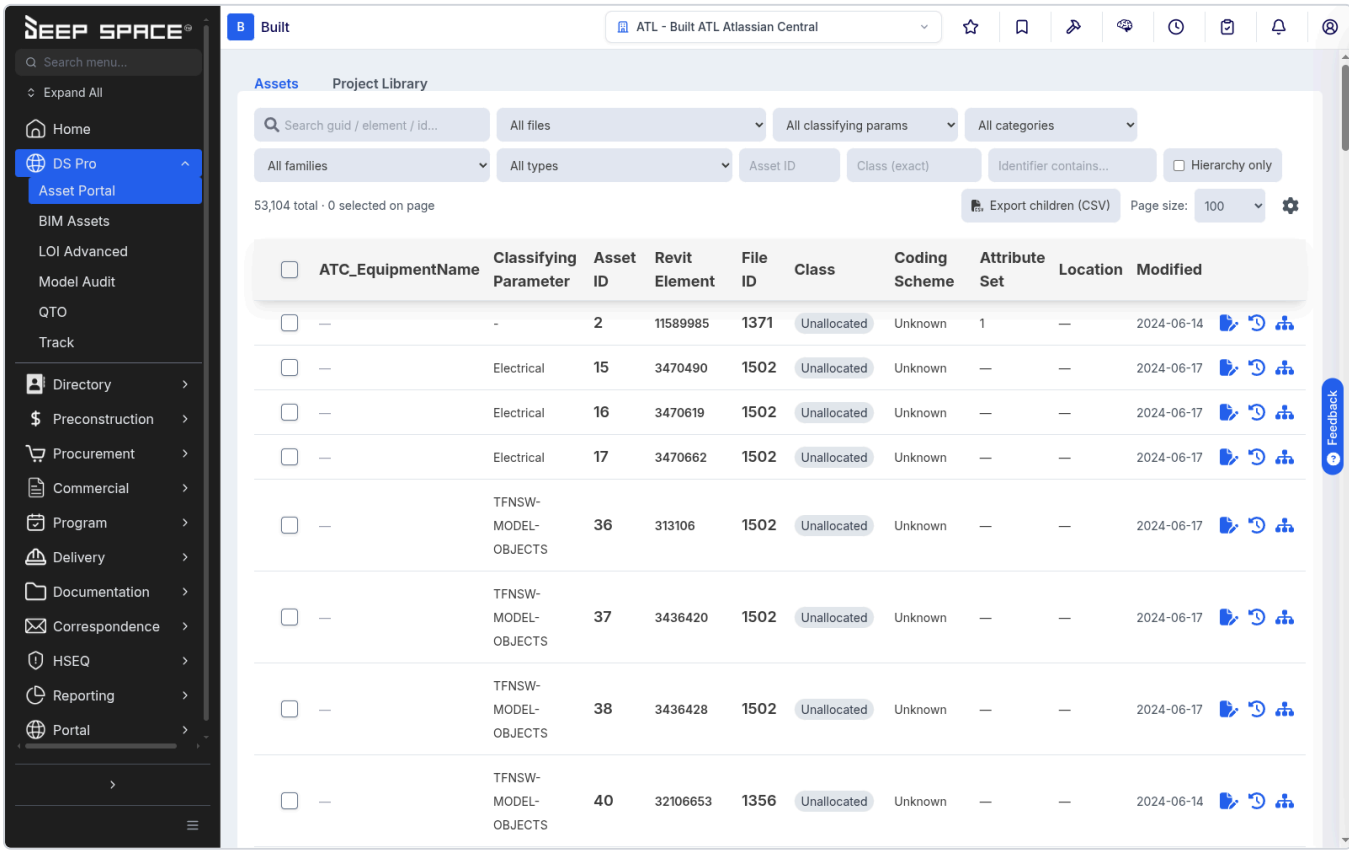
The Report table gives, per **file, category and required attribute**: the quantity of elements, the number **with data**, the number **blank** and the **% compliant (valid)**. It is the table to paste into a progress report, or to export with **Generate PDF**.

Part 4: The asset register in the Asset Portal

The Asset Portal is the register itself: every element that the post-sync pipeline has made an asset (Part 2), with its asset data, and the **child assets** that trades add for equipment that is not modelled.

Open Horizon, select the project, then **DS Pro ▸ Asset Portal**.

Finding assets

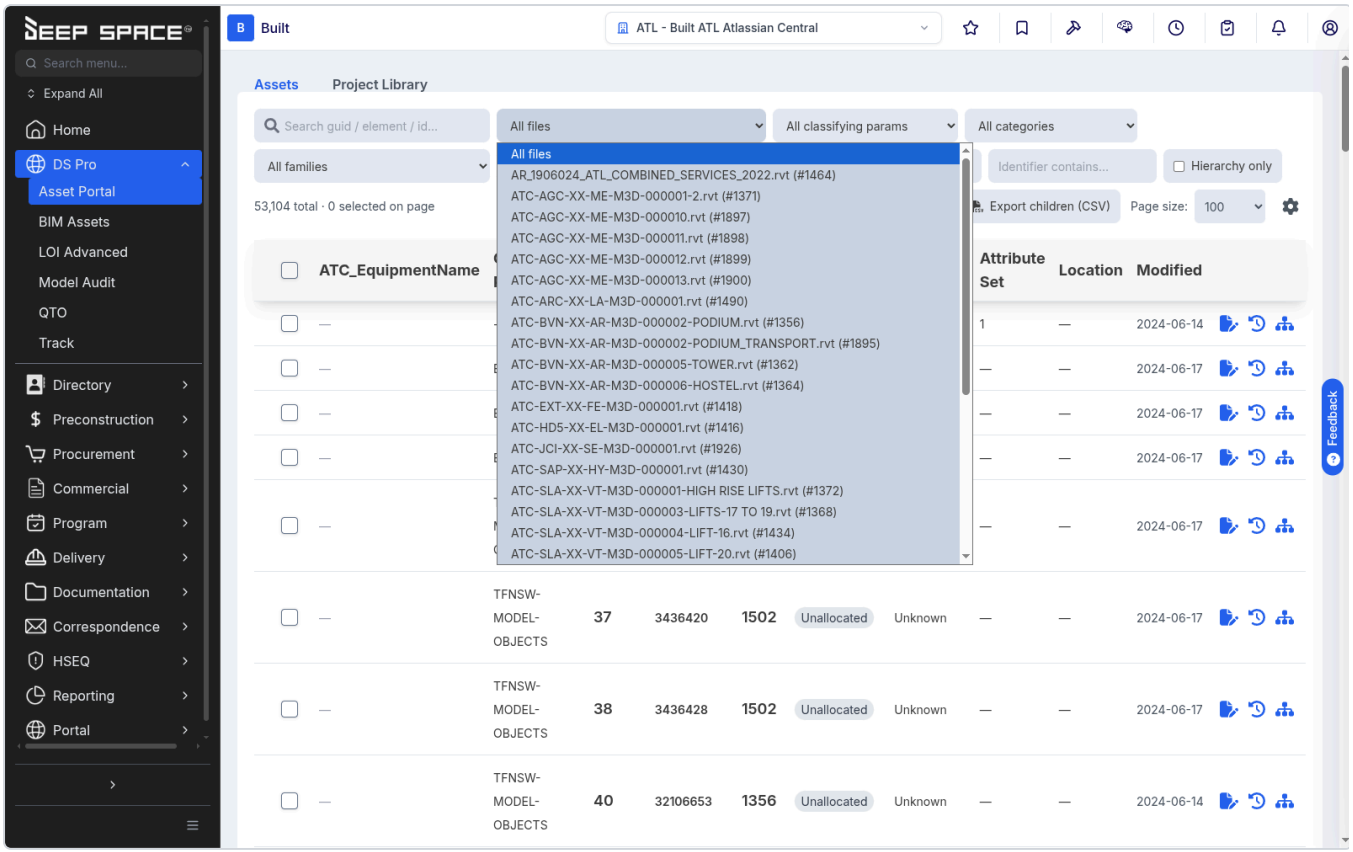


Asset Portal: the asset list with its total, filters and columns.

The **Assets** tab lists every asset, with the total at the top. Narrow it with:

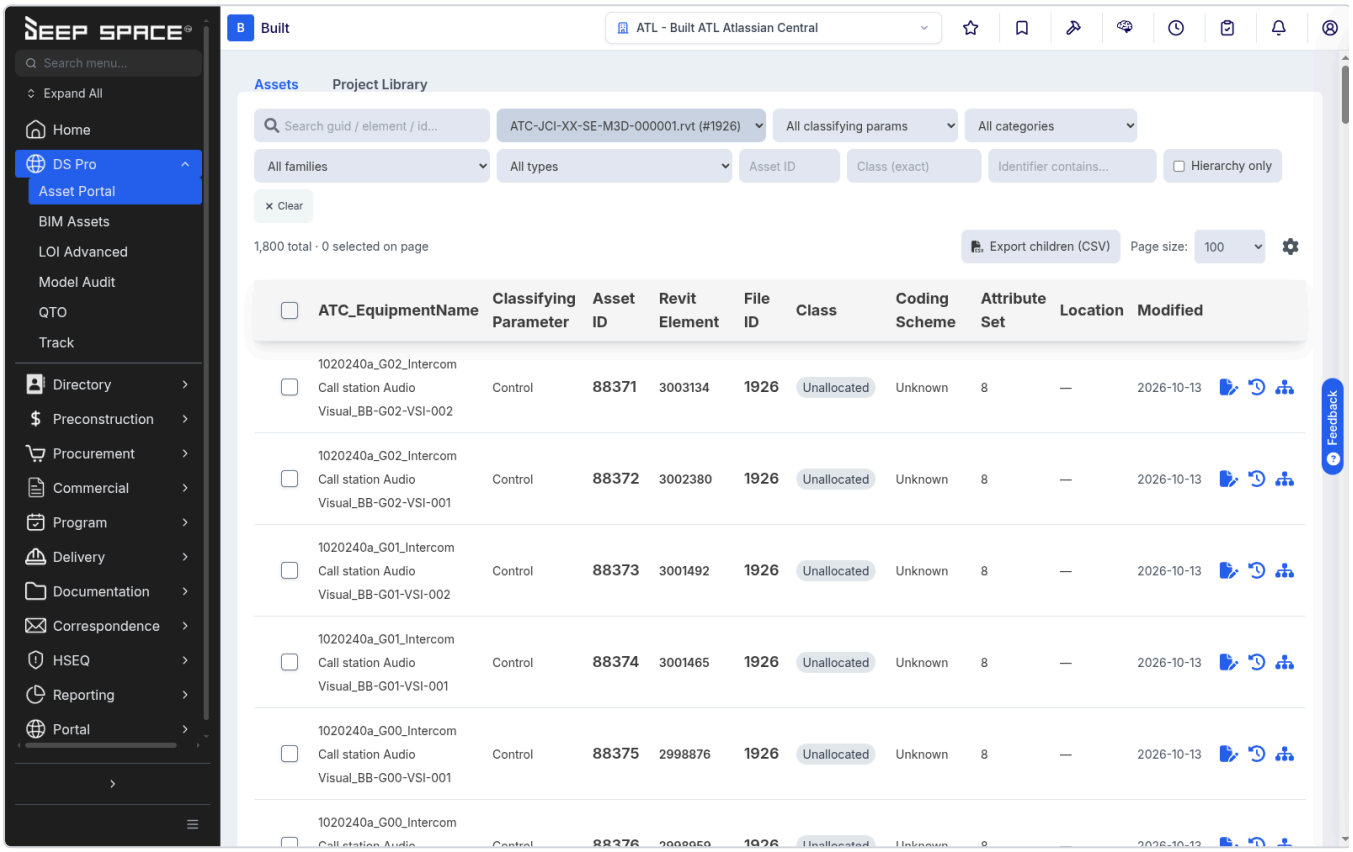
- the **search box** (GUID, element ID or asset ID);
- **file** (one model), **classifying parameter**, **category**, **family** and **type**;
- **Asset ID**, **Class (exact)** and **Identifier contains…** (part of the Equipment Name);
- **Hierarchy only**, to show parents that have children.

Each row shows the **Equipment Name** (for Built, `ATC_EquipmentName`), classifying parameter, asset ID, **Revit element ID**, file ID, class, coding scheme, attribute set and location. The icons on the right open the asset's data, its change history, and its children.



The file filter lists every model that has assets.

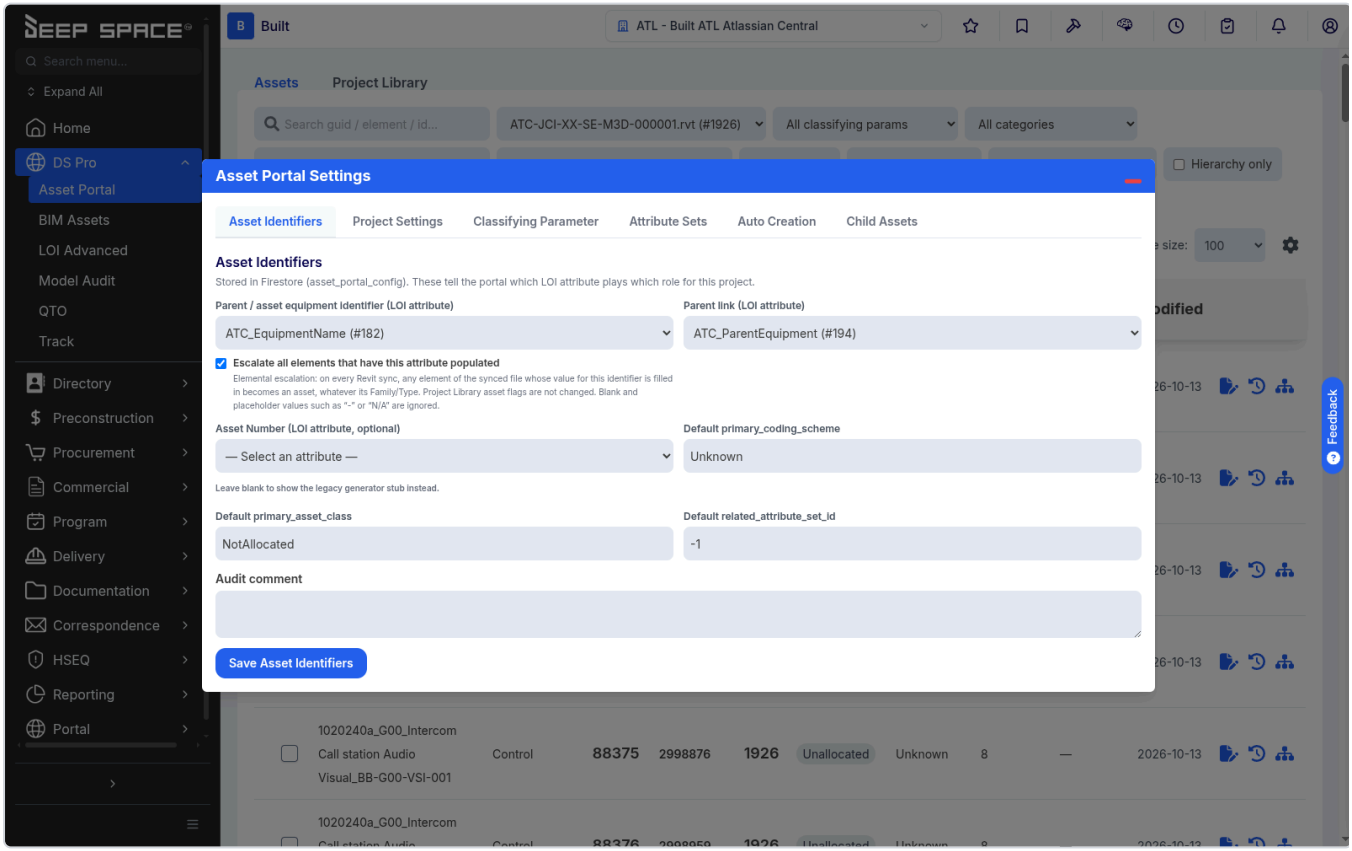
The **file** filter lists only models that have at least one asset. If a model you expect is missing, it has either not synced since asset escalation was switched on, or it has no element that qualifies as an asset (Part 2).



The security model's assets, filtered by file.

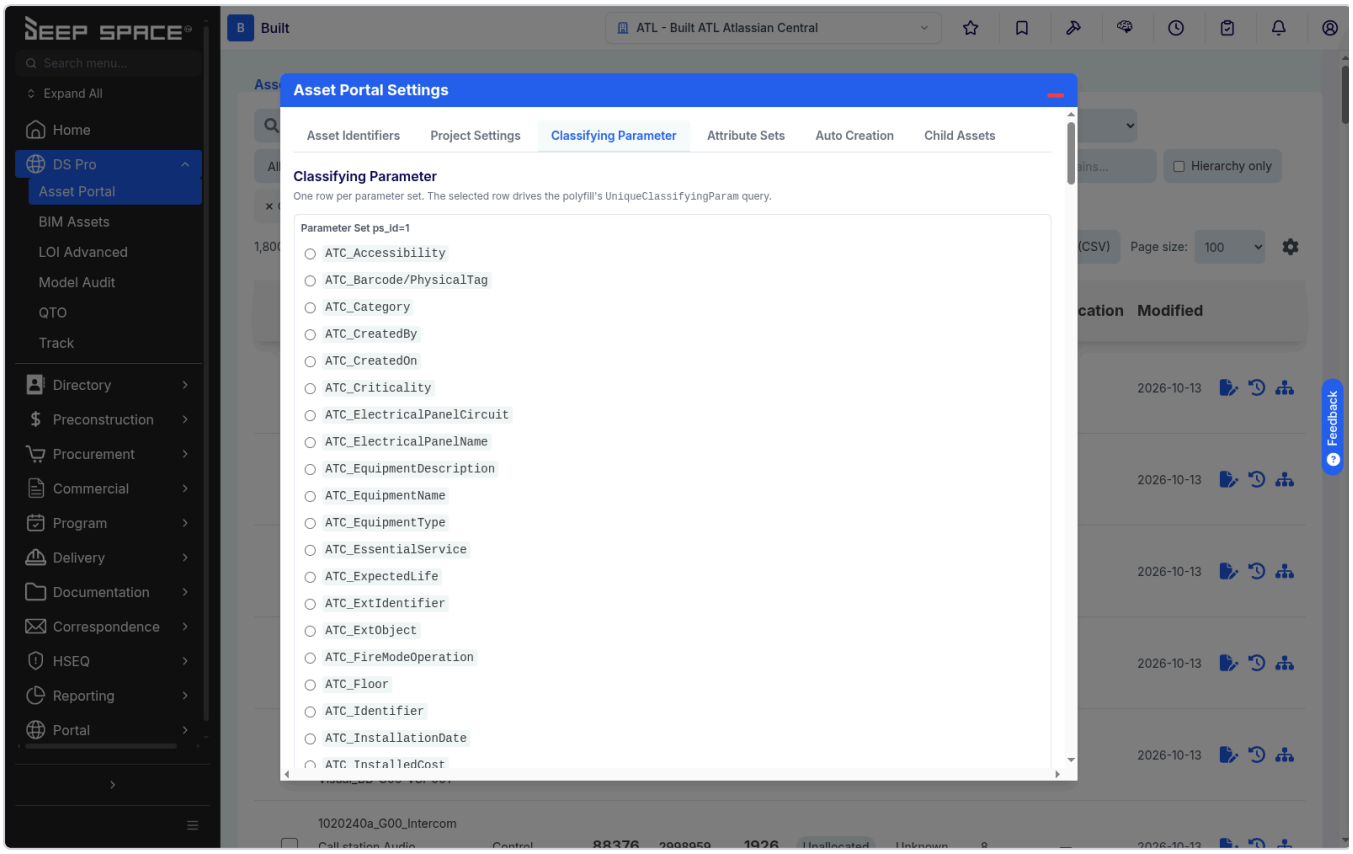
Asset Portal settings (project managers)

The gear icon opens **Asset Portal Settings**. These settings control how assets are created and how child assets work. Only project managers can change them, and every change is recorded with an audit comment.



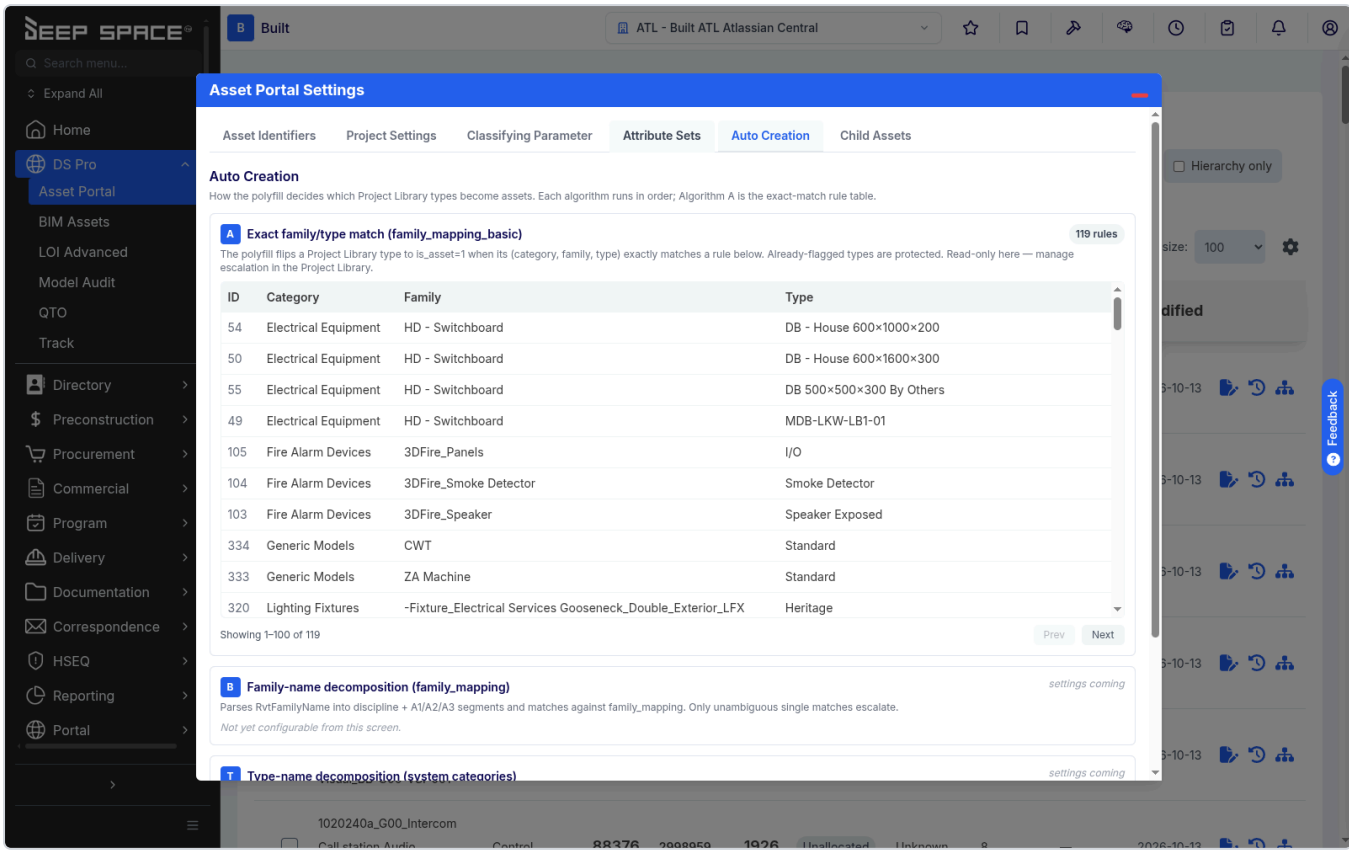
Asset Identifiers: the Equipment Name attribute, the parent link, and the elemental escalation switch.

- **Asset Identifiers.** Which LOI attribute is the asset's **equipment identifier** (Built: `ATC_EquipmentName`) and which is the **parent link** for children (Built: `ATC_ParentEquipment`). The tickbox **Escalate all elements that have this attribute populated** switches on elemental escalation (Part 2): every element whose Equipment Name is filled in becomes an asset on every sync.



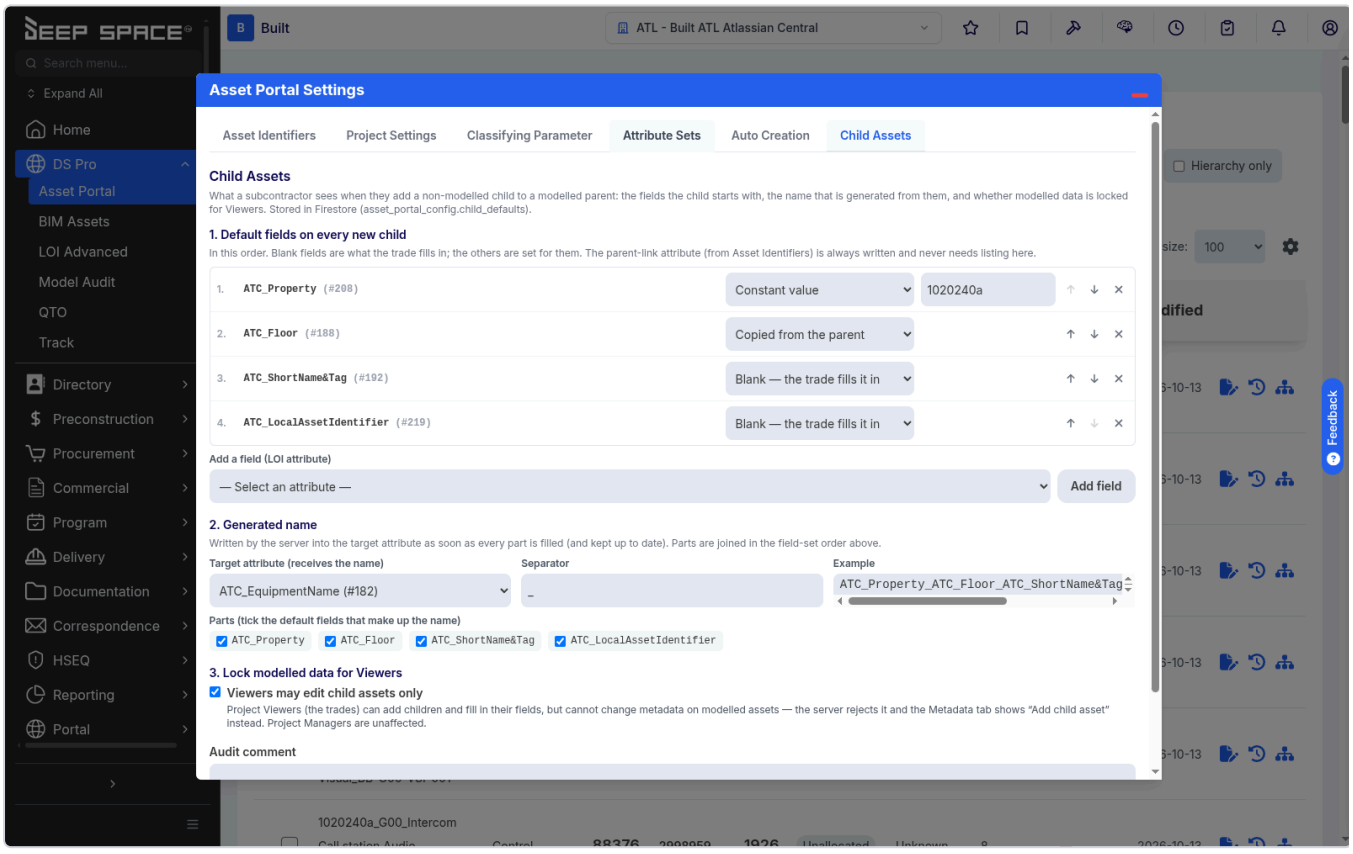
Classifying Parameter: one row per parameter set; the selected parameter decides each element's attribute set.

- **Classifying Parameter.** The parameter whose value chooses each element's attribute set (Built: `ATC_Category`).
- **Attribute Sets.** The attribute sets and their members (the same requirements LOI Advanced checks).



Auto Creation: the family/type rules that turn whole types into assets.

- **Auto Creation.** The **family/type rules** that make every element of a matching Revit category, family and type an asset, whether or not it has an Equipment Name (Part 2).



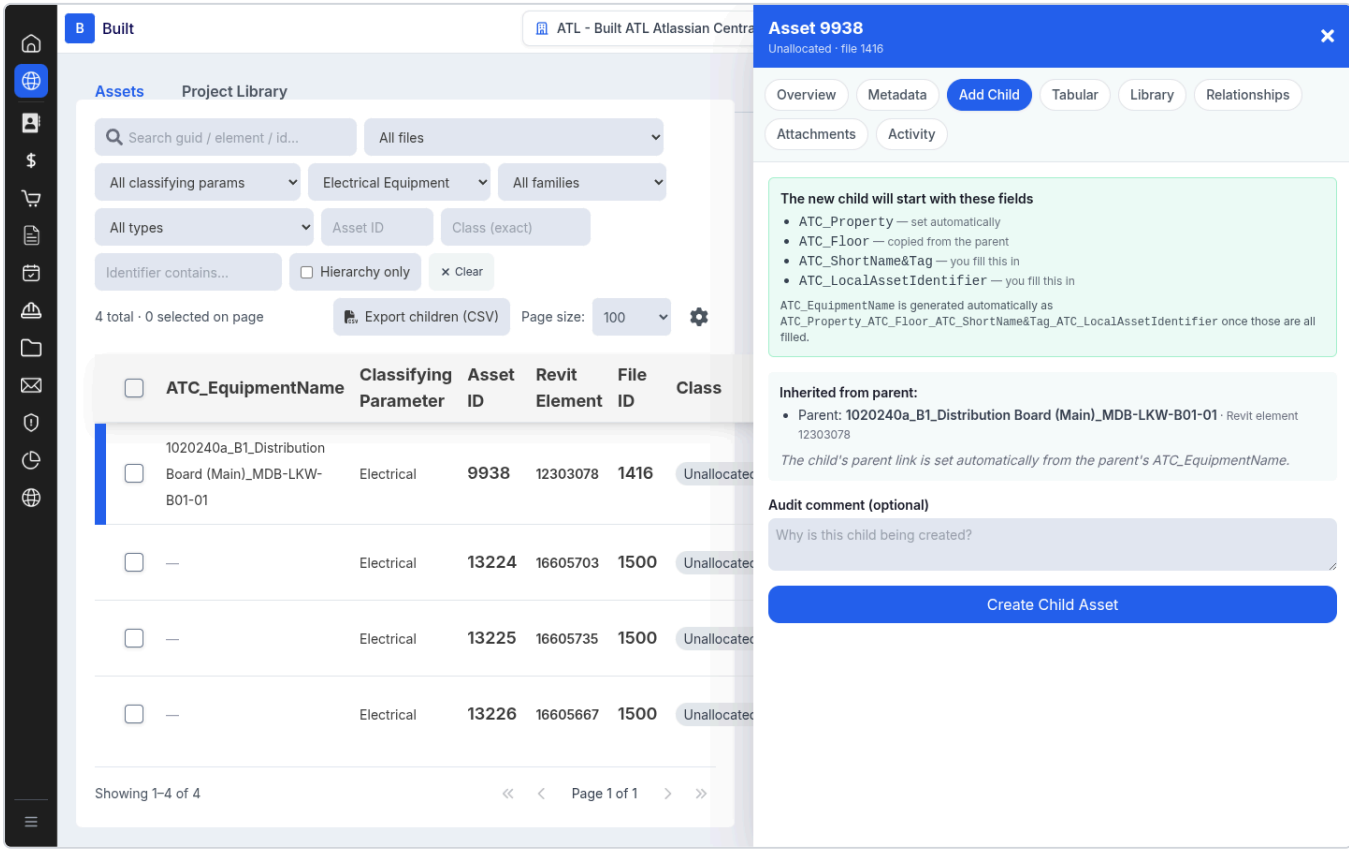
Child Assets: default fields, the generated Equipment Name, and the lock on modelled data.

- **Child Assets.** The fields every new child starts with, how the child's Equipment Name is generated from them, and whether trades are locked out of modelled (parent) data.

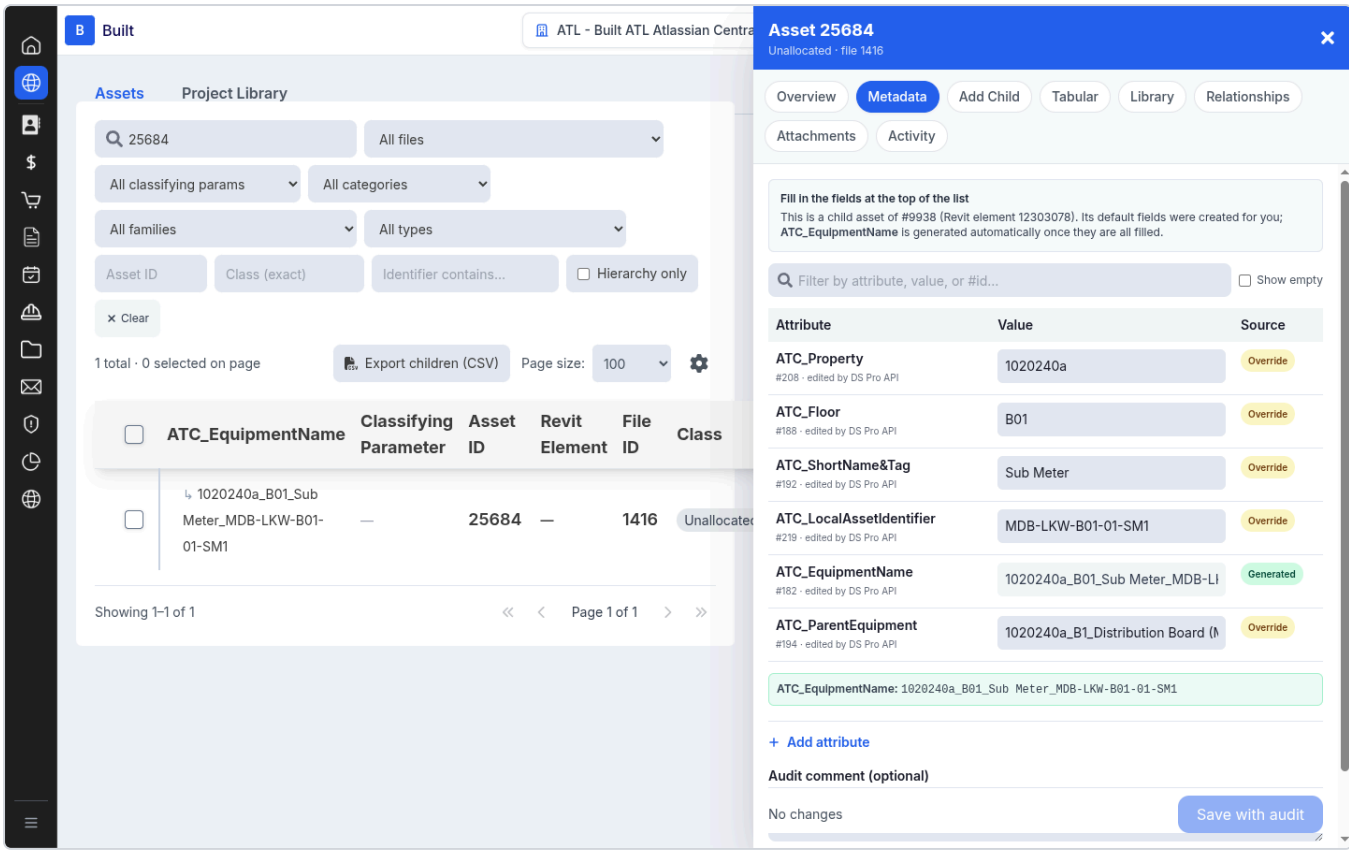
Child assets (non-modelled equipment)

Trades add **child assets** for equipment that is not modelled in Revit (for example the components inside a switchboard), under the modelled parent:

1. Find the parent asset (filter by category, family and type, or search by the Revit element ID).
2. Open **Add Child**, create the child, and fill in the blank fields.
3. The child's Equipment Name is generated automatically when every part is filled in.



Adding a child under a modelled parent.



Filling in the child's fields.

The full step-by-step for trades is in the **Asset Portal child assets guide**: https://cdn.deepspacesync.com/help-guides/ASSET-PORTAL-CHILD-ASSETS-GUIDE.pdf

Export children (CSV) downloads the children for the current filter, including the parent's Revit element ID, ready to paste into the asset register or to push back into Revit with your own tools.

Putting it together: the routine

After every model update

1. The model is published on ACC.
2. The scheduled sync picks it up. Check **LOI Advanced ▸ Model Register** (after a refresh) or the Asset Portal file filter.
3. Select **Refresh Dataset** in LOI Advanced.
4. Work through **Overview** → **Compliance** → **Details**, filtered by package or discipline, and send each trade its list of *Does Not Exist*, *Blank* and *Invalid* items.
5. Trades fix the data in Revit and publish; the next sync updates both apps.
6. New assets appear in the Asset Portal automatically; trades add children where needed.

Roles

Role	Routine
Modeller or trade	Fill in the required asset parameters in Revit, including `ATC_Category` (which decides what is required) and `ATC_EquipmentName` (which makes the element an asset). Add child assets in the Asset Portal
Digital engineering coordinator	After each sync, refresh LOI Advanced, review Compliance and Report, and chase the gaps by trade. Spot-check new assets in the Asset Portal
Project manager (Asset Portal)	Keep the Asset Portal settings current: identifier, escalation switch, child defaults. Archive test or mistaken children
Deep Space	Run and monitor the syncs, keep the parameter whitelist in step with the LOI requirements, and maintain the pipeline

Troubleshooting

Symptom	What to check
A model is missing from the Asset Portal file filter	It has no assets yet. Check it has synced since escalation was switched on (Model Register), and that its elements have a filled-in Equipment Name or match a family/type rule
A value fixed in Revit still shows as wrong	Was the model published and synced since? Then **Refresh Dataset** in LOI Advanced
A parameter shows **Does Not Exist** everywhere, but it is filled in in Revit	Is it on the parameter whitelist? A new LOI parameter needs a whitelist update and a re-sync (ask Deep Space). Also check the parameter name is spelled exactly the same, including capitals and spaces
A model shows **Not Synced** but you know it synced	It was probably skipped as unchanged, so it has no new data in the recency window. Its data is still current
An element with an Equipment Name is not an asset	The value may be a placeholder (blank, "-", "N/A", "TBC" …), which is ignored. Or the model has not synced since the name was added
An asset has no attribute set (shows -1)	Its classifying value (for example "Mechanical, Filtration Unit") does not match any attribute set name. Correct `ATC_Category` in Revit, or ask for the attribute set to be added
An asset disappeared	Its element was deleted from the model (or replaced, which gives it a new id). Assets are removed when their element leaves the model
LOI Advanced numbers look old	Check the refresh time on the Overview tab, then select **Refresh Dataset**

Quick reference

Task	Where
See when each model last synced	LOI Advanced ▸ Model Register

Task	Where
Update the data checks after a sync	LOI Advanced ▸ Refresh Dataset
See what each kind of element needs	LOI Advanced ▸ LOI Requirements
Find one element's problems	LOI Advanced ▸ Details ▸ Element ID Search
Progress by file and attribute	LOI Advanced ▸ Report
Export for facility management	LOI Advanced ▸ Data Export ▸ FM Export
Find an asset	Asset Portal ▸ Assets (search or filters)
Change how assets are created	Asset Portal ▸ Settings ▸ Asset Identifiers / Auto Creation
Add equipment that is not modelled	Asset Portal ▸ asset ▸ Add Child
Export children for the register	Asset Portal ▸ Export children (CSV)

Related guides

- Asset Portal child assets guide (for trades): https://cdn.deepspacesync.com/help-guides/ASSET-PORTAL-CHILD-ASSETS-GUIDE.pdf
- How Contractor Compliance Works in Deep Space (company compliance, a separate topic): https://cdn.deepspacesync.com/help-guides/How%20Contractor%20Compliance%20Works%20in%20Deep%20Space.pdf

Questions or a model that is not syncing: contact Deep Space support.