ElectroNet

Westpower

MICROSOFT CLOUD → SNIPE-IT

The Snipe-IT integration project

Automated user and asset sync from Entra and Intune, orchestrated by FME middleware.

Entra | Intune | FME | Snipe-IT

ElectroNet

Westpower

WHO AM I?

Why I'm presenting today

Software and systems engineer, 35+ years in software development.

Transformer-based tools are not new to me — Unity3D Bolt and Unreal Engine Blueprints work much the same way.

Today I'm showing how I solved a problem in FME Transformer flow that I would otherwise have written in code.

ElectroNet

Westpower

WHERE I STARTED

Learning FME from zero

No FME experience — total beginner

Learned by building: tutorials, the Safe community forum, and a lot of failed runs.

My first integration using a no-code environment.

A course with Locus got me off the ground — big shout out to Gavin.

Leaned on AI tools — Claude and ChatGPT — to work things out.

ALWAYS LEARNING

Building my skills in FME

By the end, I had a solid understanding of FME work flow.

ElectroNet

Westpower

THE PROJECT

What the project is

01

Integration of Microsoft into Snipe-IT asset management.

02

Automated creation of users and assets — computers, iPads and iPhones.

03

Reporting on leavers who still have assets assigned.

04

Reporting on users holding more than one type of device.

ElectroNet

Westpower

THE CAST

What are they?

Entra

Microsoft's identity directory. Every staff member, and the groups they belong to.

WHO YOU ARE

Intune

Microsoft's device management. Every managed laptop, iPad and iPhone, and who signs into it.

WHAT YOU HOLD

Snipe-IT

Our asset register. The single record of what we own and who has it.

THE REGISTER

FME The glue. It reads Entra and Intune, works out what changed, and writes to Snipe-IT.

THE DESTINATION

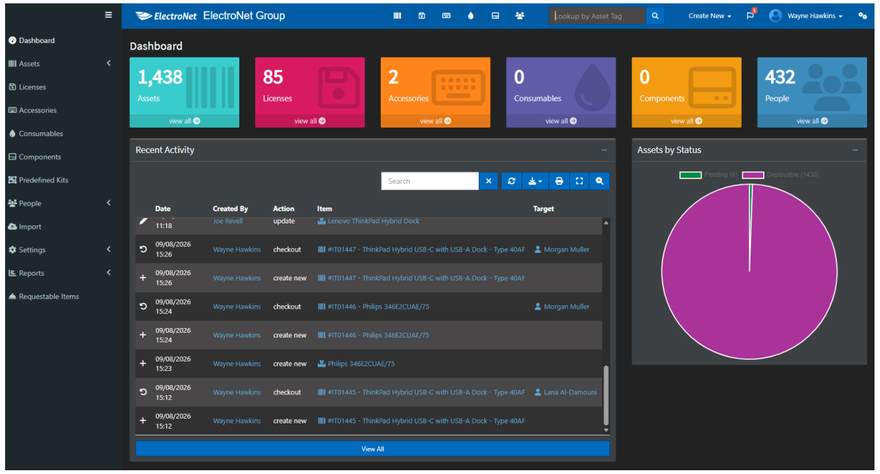
A closer look at Snipe-IT

Open-source asset management. Every device (screens, docks, laptop, iPad and iPhone) is recorded with a tag, model and status, and assets are checked out to people. Allows us to see how old devices are, if they're in warranty and helps with budgeting future costs for hardware.

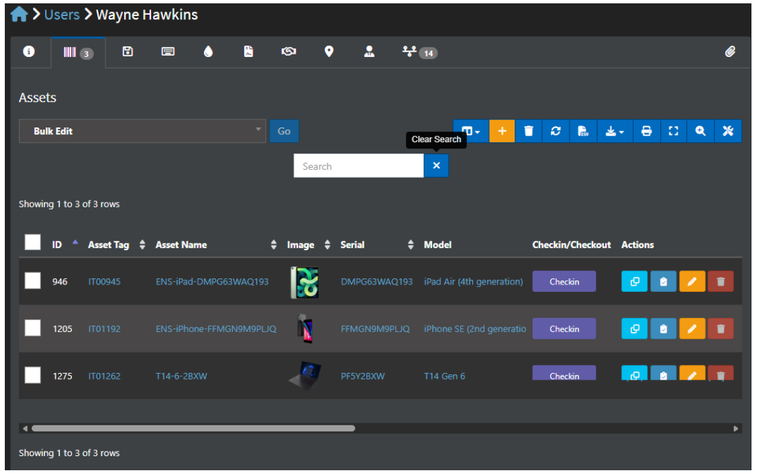
1,438 assets

432 people

DASHBOARD



ASSETS CHECKED OUT TO ONE USER



THE API Everything on these screens is reachable over the Snipe-IT REST API — which is how FME writes to it.

ElectroNet

Westpower

THE BRIEF

Two key objectives

01

Must be built on FME

Developing in code is no longer allowed.

02

No database

Everything must be API calling in real time. FME does all the heavy lifting.

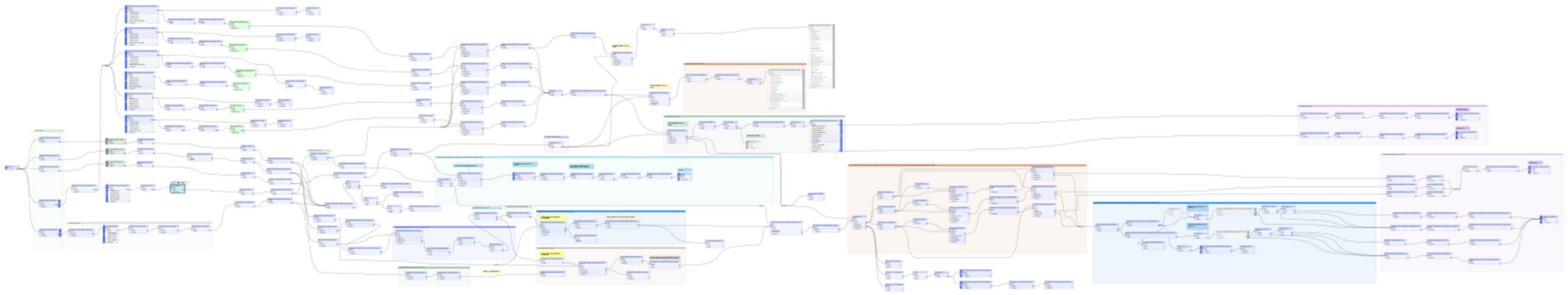
Both constraints shaped every decision that follows.

ElectroNet

Westpower

ALL OF THE LOGIC

It took a lot of Transformers



ElectroNet

Westpower

IMPORT OF DEVICES — PART 1

Getting devices out of Intune

Step 1

Authenticate with Microsoft Graph

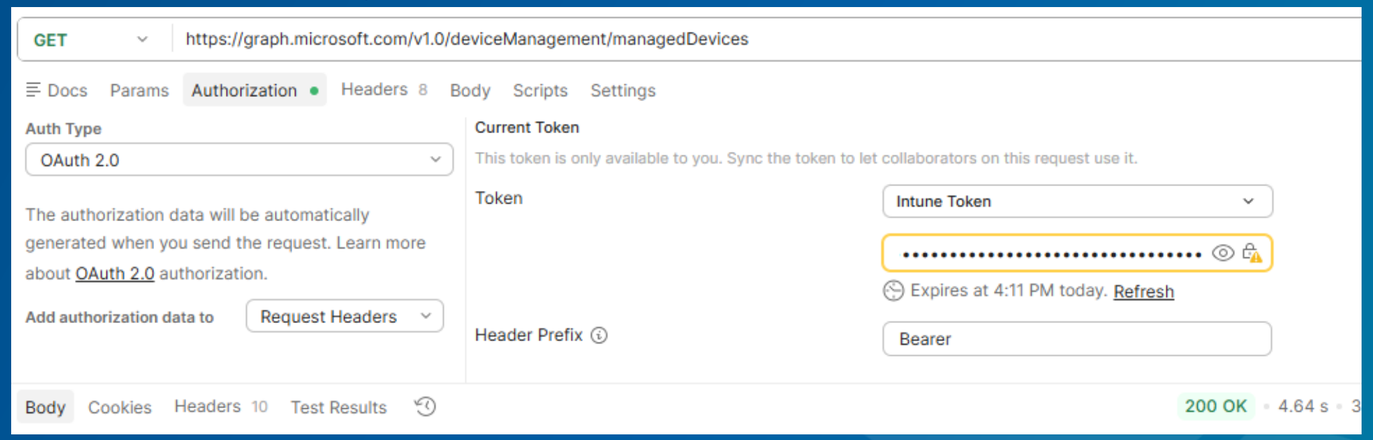
OAuth 2.0 call returns a bearer token, held for the life of the run.

Step 2

Request every device as JSON

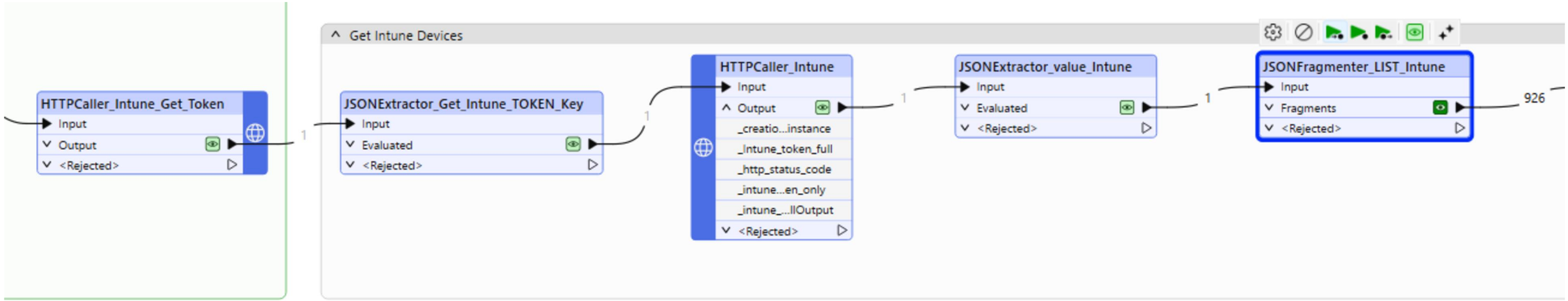
GET /v1.0/deviceManagement/managedDevices, then fragment the list — 926 devices in this run.

THE GRAPH CALL, PROVEN IN POSTMAN



STEPS 1 AND 2 IN FME

The whole Intune read, five transformers



01
Call Graph for a token

02
Extract the token key

03
Call managedDevices with it

04
Extract the value array

05
Fragment into 926 devices

ElectroNet

Westpower

IMPORT OF DEVICES — PART 2

Step 3 — does it already exist?

CHECK

Does the asset exist in Snipe-IT?

No — Create the asset in Snipe-IT and check it out to the user.

Yes — **Is it still assigned to another user?**

- **Not assigned** Check it out to the new user.
- **Still assigned** Notify IT — the asset needs checking in first.

Nothing is reassigned silently. A device held by someone else becomes a job for IT, not an automatic change.

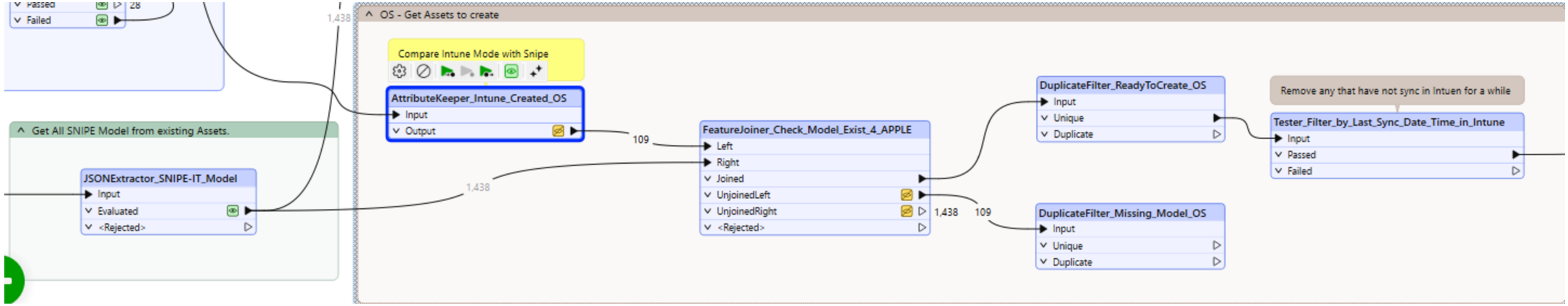
ElectroNet

Westpower

BEFORE AN ASSET CAN BE CREATED

The model has to exist first

Snipe-IT will not accept an asset without a model, for example "T14-5-xxx Laptop". Every Intune device model is checked against the models already in Snipe-IT before anything is created.



Join
Snipe-IT's models are matched against the device model name.

Matched
Ready to create against a known model.

Missing
Held back, and IT is emailed to say a new model type is missing.

ElectroNet

Westpower

SAME IN CODE

794 lines of code

INTUNESNIPESYNC.CS scroll ↕

```
using System.Net;
using System.Net.Http.Headers;
using System.Net.Http.Json;
using System.Text;
using System.Text.Json;
using System.Text.Json.Serialization;

namespace IntuneSnipeSync;

// ============================================================
//  CONFIG
// ============================================================

public class SyncConfig
{
    public string TenantId { get; set; } = "";
    public string ClientId { get; set; } = "";
    public string ClientSecret { get; set; } = "";

    public string SnipeBaseUrl { get; set; } = "https://electronet.snipe-it.io";
    public string SnipeApiKey { get; set; } = "";

    // Snipe-IT status label applied to newly created assets (must be a
    // "deployable" status or the checkout call will be rejected).
    public int NewAssetStatusId { get; set; } = 2;

    public string ReportFromAddress { get; set; } = "intune-sync@electronet.co.nz";
    public string ReportToAddress { get; set; } = "it@electronet.co.nz";

    // Safety valve: log what would happen without writing to Snipe-IT.
    public bool DryRun { get; set; } = true;
```

THE PART THAT MATTERS

```
// Model missing → report, stop (can't create an asset without a model)
if (model is null) { TallyMissingModel(device); return; }

// Asset missing → create, then assign
if (asset is null) {
    var created = await _snipe.CreateAssetAsync(device, model.Id);
    if (intuneUser is not null)
        await _snipe.CheckoutToUserAsync(created.Id, intuneUser.Id, note);
    return;
}

// Asset exists, unassigned → assign
if (asset.AssignedTo is null) {
    await _snipe.CheckoutToUserAsync(asset.Id, intuneUser.Id, note);
    return;
}

// Asset exists, someone else has it → report, don't touch
if (asset.AssignedTo.Id != intuneUser?.Id)
    StillInUse.Add(new InUseRow(...));
```

ElectroNet

Westpower

Getting user information from Entra

STEP 1

Get a token from Microsoft

The same authentication call as the device import — one bearer token for the run.

STEP 2

Call Graph for user details

Name, email, job title, department — everything Snipe-IT needs to hold a person.

Same two steps as Intune. What comes back is a different shape entirely.

ElectroNet

Westpower

THE CATCH

Entra does not give you everyone at once

INTUNE

One call, 926 devices

The whole list arrives in a single response, ready to fragment.

ENTRA

30 users, plus a URL

Each response carries 30 users and a link to the next 30. Follow it, or you only ever see the first page.

WHAT COMES BACK — THE URL WE HAVE TO FOLLOW

@ODATA.NEXTLINK The URL for the next page. When it stops appearing, you have everyone.

ElectroNet

Westpower

A loop calls a method, the method answers

THE LOOP

`while url is not null`
Start with the first URL.

`(nextUrl, dataset) = doWork(url)`
Hand it the URL and wait for both objects back.

`keep dataset, test nextUrl`
Keep the data, then go round again on the URL.

↑ Back to the top until there are none left

`url`
call, with a URL

`(nextUrl, dataset)`
return, a tuple of two

THE METHOD

`doWork(url)`

Run the steps — the work happens here, once.

Same steps every call, only the value changes.

Work out both results — the data, and where to go next.

`return (nextUrl, dataset)`
One way out, returning a tuple — two objects in a single answer.

THE POINT The method returns a tuple — the next URL and the dataset — and never knows how many times it runs. The loop decides that.

ElectroNet

Westpower

IF I HAD WRITTEN IT

The same loop, in code

WHAT THAT LOOKS LIKE WRITTEN OUT

```
while (!string.IsNullOrEmpty(url))
{
    (string? nextUrl, List<DTO_Users> users) = await GetPageAsync(url);
    allUsers.AddRange(users);
    url = nextUrl;                    // null on last page -> loop exits
}
```

`while (url != null)`

Keep going while there is a next page.

›

`GetPageAsync(url)`

The method calls the API and hands back the users and the next URL.

›

`allUsers.AddRange(users)`

Add this page to the full set.

›

`url = nextUrl`

Reassign, then test again at the top — null on the last page and the loop exits.

ElectroNet

Westpower

THE SOLUTION

A 'Custom Transformer' that loops on itself

01
Call the URL and keep the 30 users it returns.

›

02
Test for a 200, then pull out the users and the next URL.

›

Yes
A next URL came back — the Looper feeds it to the input and runs again.

›

No
Exit with the full set — 1,491 users in this run.

THE THREE PORTS FME GIVES YOU

The loop port is what makes a transformer able to call itself.

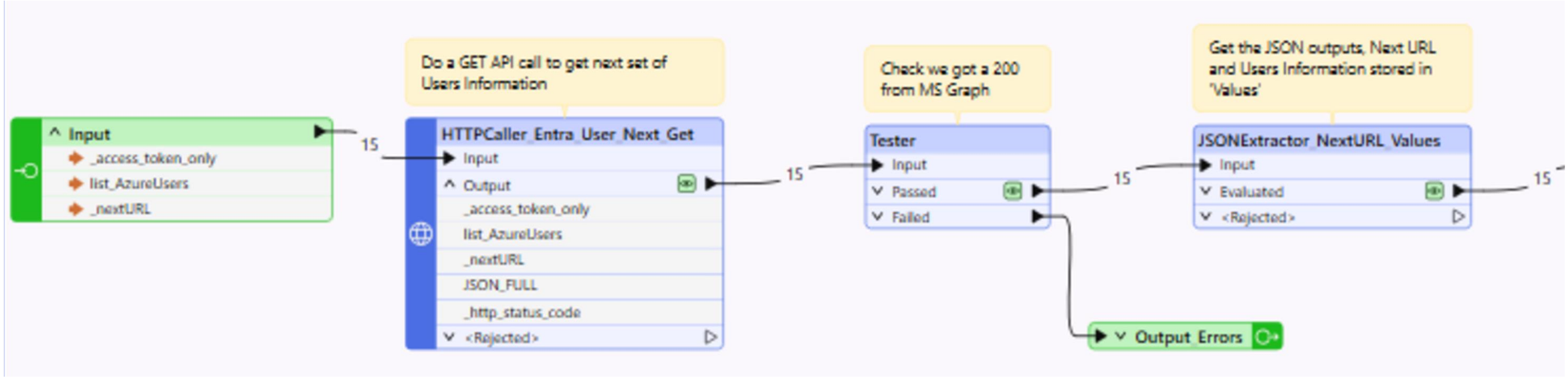
Written once as a 'Custom Transformer', so any workspace that needs Entra users can reuse it.

ElectroNet

Westpower

THE FME BUILD — PART 1

Call the next page



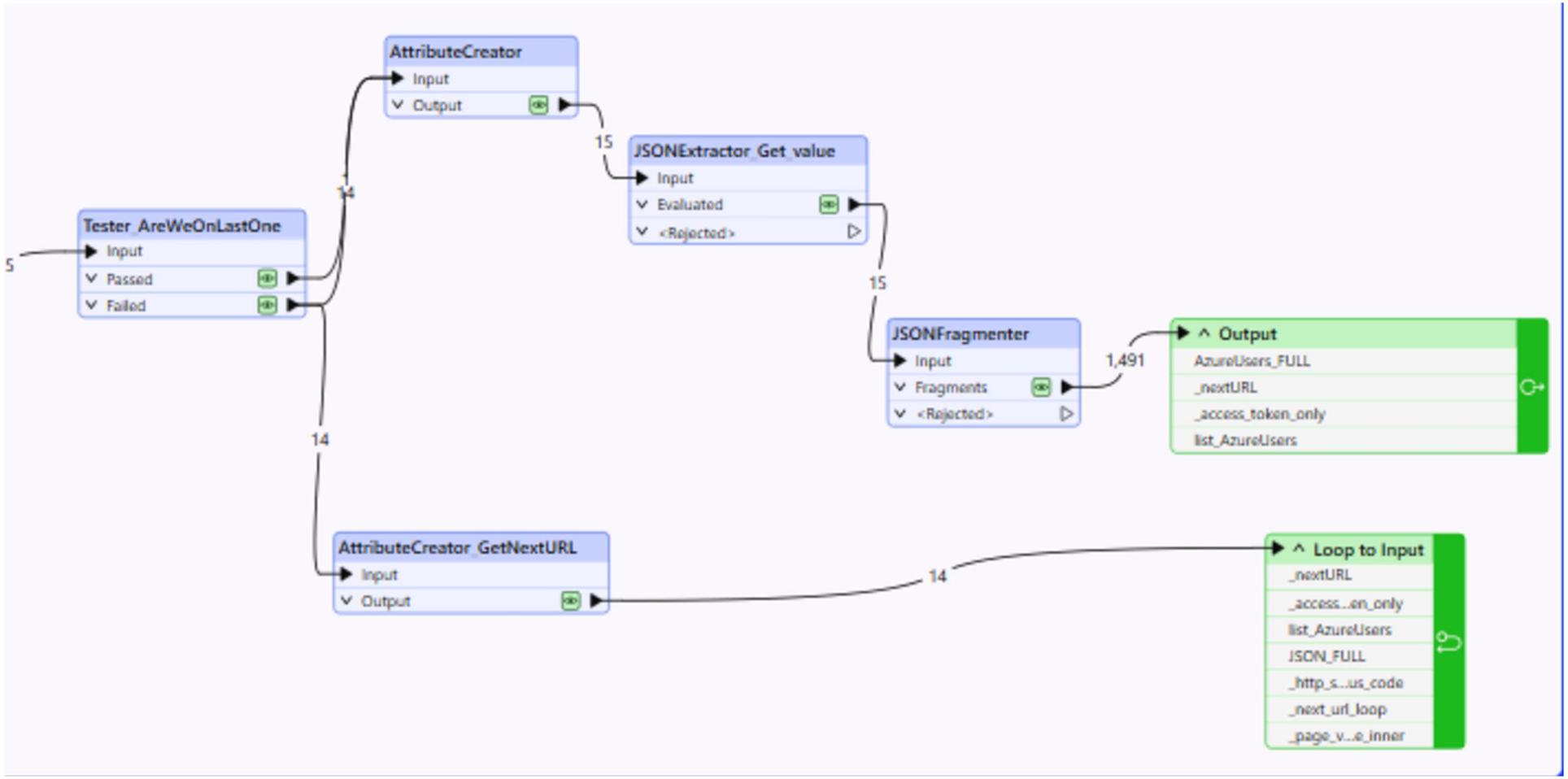
ANYTHING ELSE A failed call goes out the error port rather than stopping the run.

ElectroNet

Westpower

THE FME BUILD — PART 2

Loop back, or exit



1,491 USERS Fragmented out of the transformer once the last page comes back.

ElectroNet

Westpower

THE SNIPE-IT API

Adding and updating in Snipe-IT

01

Add users

People come across from Entra before anything can be assigned to them.

`POST /api/v1/users`

02

Create asset models

No asset can exist without its model, so missing models are created first.

`POST /api/v1/models`

03

Create assets

Each Intune device becomes an asset with a tag, model and status.

`POST /api/v1/hardware`

04

Assign and update

The asset is checked out to its user, or updated if it moved.

`POST /api/v1/hardware/{id}/checkout`

The order matters. Each step depends on the one before it existing in Snipe-IT.

ElectroNet

Westpower

LEAVERS

Exit logic

01

Removed from the group

The only signal we need. When someone leaves, they drop out of the watched Entra group.

›

02

FME sees they have left

The next run compares group membership and flags the person as gone.

›

03

We get notified

An email lists what is still checked out to them, so we can gather all the equipment.

NOTHING IS DELETED

The assets stay checked out until someone physically has them back. The IT team then returns them in Snipe-IT — we don't allow FME to do this.

ElectroNet

Westpower

WORKING THE PROJECT

Building rules so others can follow your work

Rules so somebody else can open this workspace and work on it.

A workspace nobody else can read is a workspace only I can maintain.

ElectroNet

Westpower

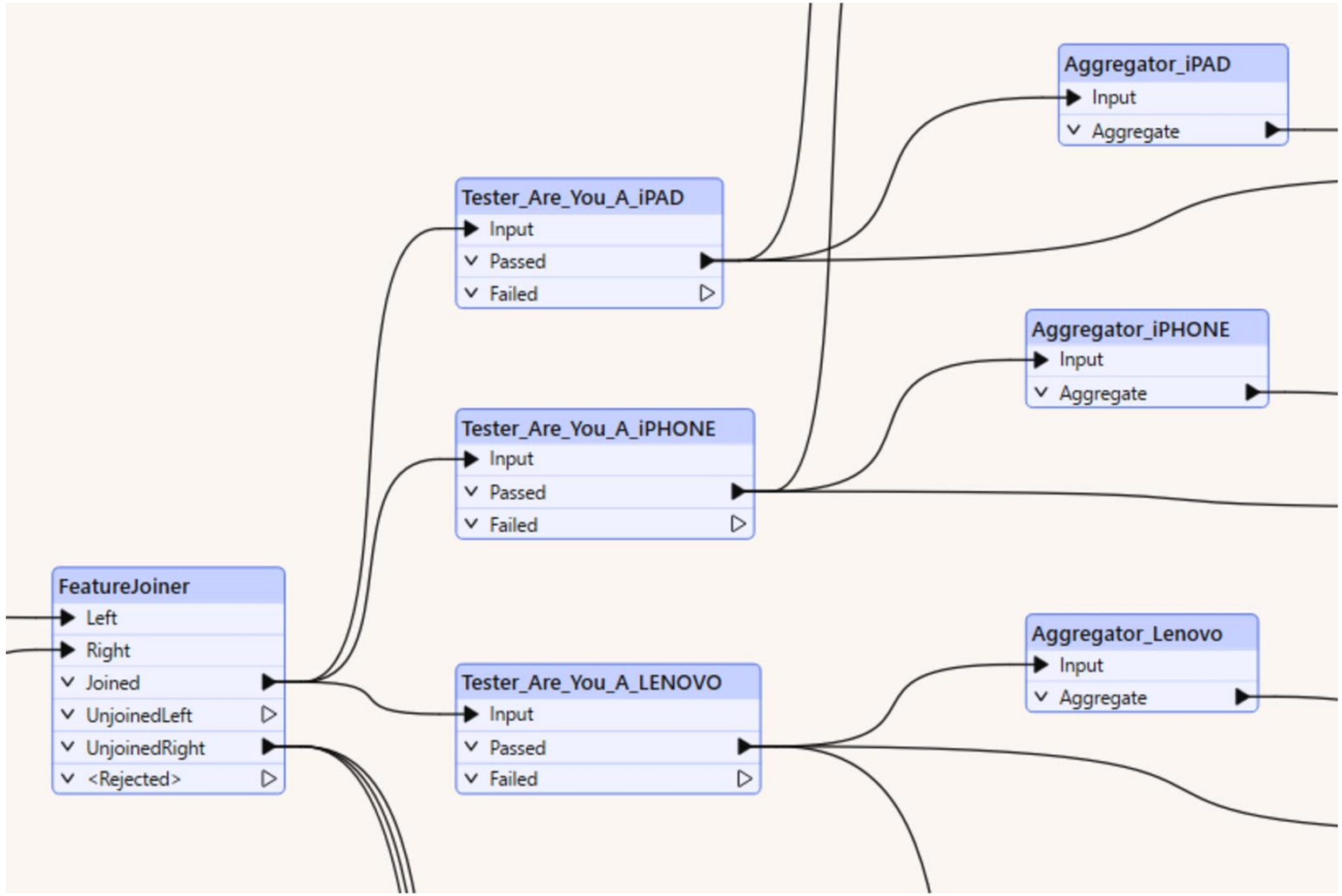
RULE 1

Always name your Transformers clearly

Keep the name of what the Transformer is, followed by a short description of what it does.

> `Tester_Are_You_A_iPAD`
>
> The type, then the question it asks.

> `Aggregator_iPHONE`
>
> You can follow the flow without opening a single Transformer.



ElectroNet

Westpower

RULE 2

Give your variables names you understand

Prefix every attribute so you know what it holds and where it came from.

`SNIPE_serial`

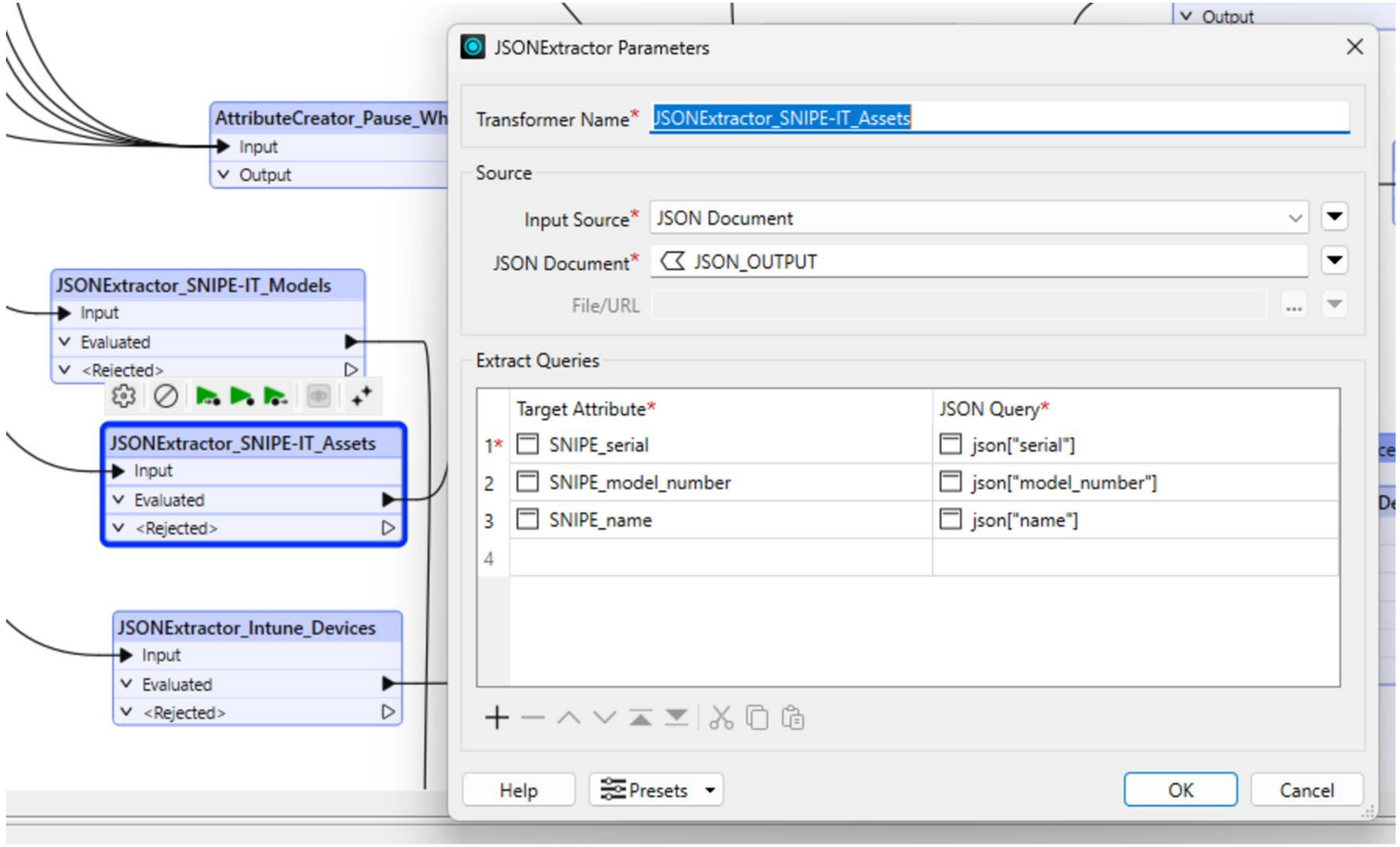
System, then the field it holds.

`SNIPE_model_number`

Reads clearly next to the JSON query beside it.

`SNIPE_name`

You never have to guess which system a value came from.



ElectroNet

Westpower

RULE 3

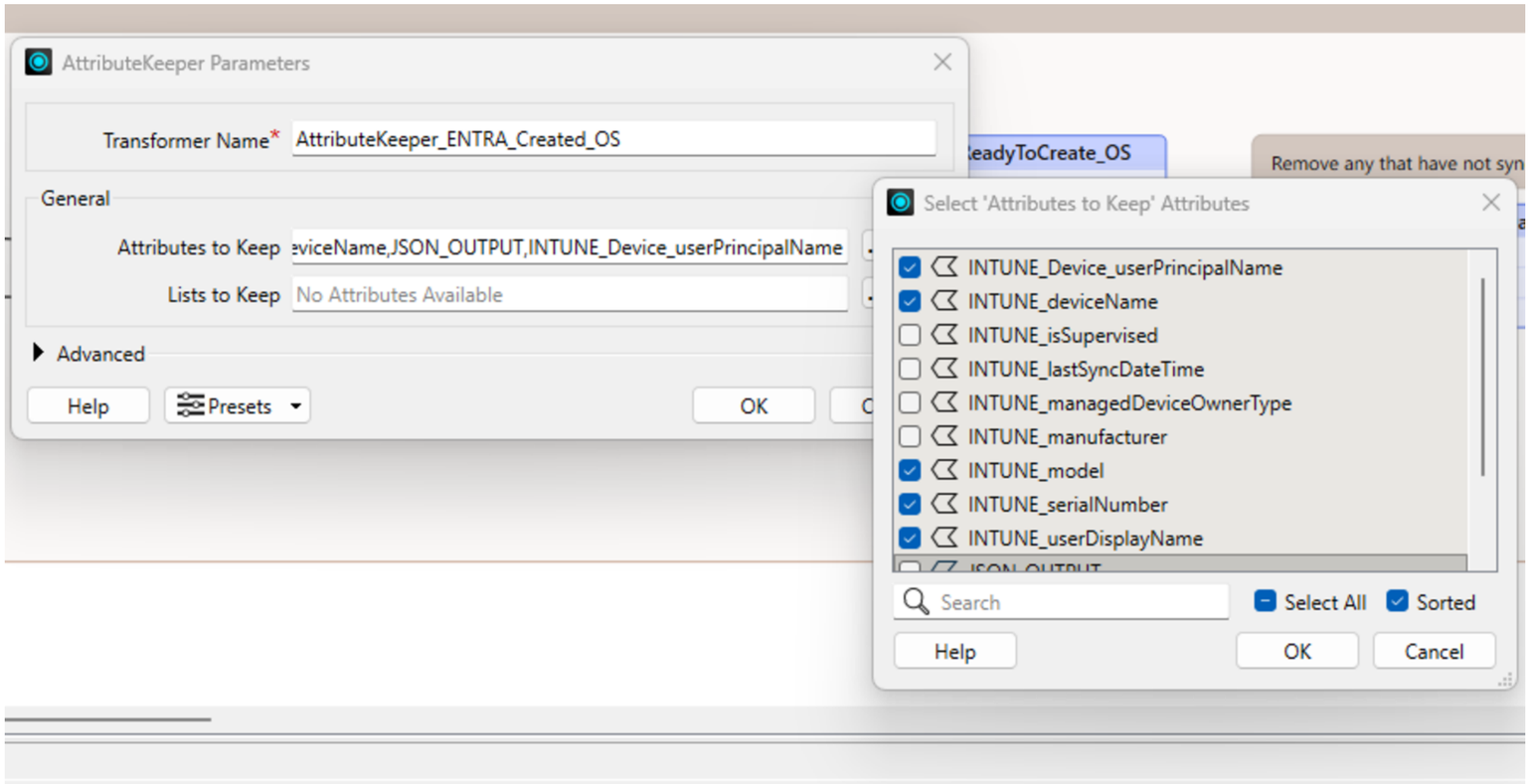
Keep it simple — drop information you know you don't need.

AttributeKeeper

Tick the handful you need — everything else is gone from that point on.

Less to read

Fewer attributes downstream means less to scroll, less to debug.



ElectroNet

Westpower

WRAPPING UP

Final opinion

01 Transformer-based scripting is not a new concept to me, so I was quick to adopt the FME style of working.

02 Once you understand that everything is large rows of data in a table, manipulating it becomes easy.

03 You don’t need to know a lot to do a lot — fewer than 12 Transformers does 95% of the work.

04 Is coding faster than Transformer-based building? Yes, because we have AI.

05 But all the FME documentation sits on a public site, so AI knows it well. ChatGPT is your friend.

OVER TO YOU

Q & A

Thanks for listening.