FME in the Forest

John Cannon

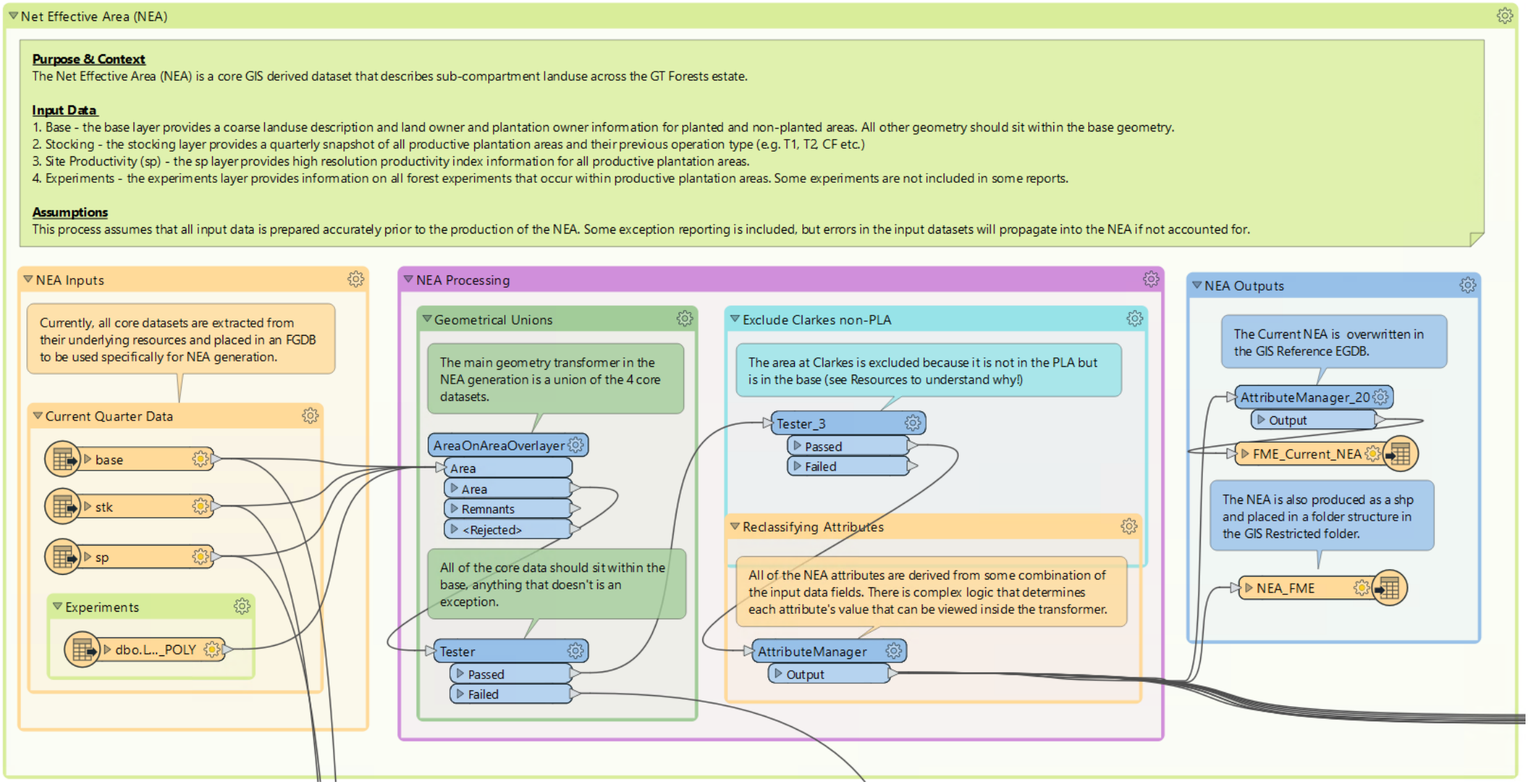
the creative fibre group

One Forty One

```
#var = raw_input("Do you wish to continue? ((y)/n)")
var = sys.argv[1]
#var = 'y'
# print "you entered ", var
if var.upper() == "N" or var.upper() == "NO":
    print "Exiting script"
    sys.Exit()



# Set the necessary product code
if arcpy.CheckProduct("ArcInfo") != "Available":
    print "ArcInfo Licence is not available"
    #sys.exit("Exiting.")

arcpy.SetProduct("ArcInfo")


arcpy.env.overwriteOutput = True

# Set the workspace
#arcpy.Workspace = r"C:\Users\mackep01\OutPut"
arcpy.Workspace = sys.argv[2]

# Load required toolboxes...
#arcpy.AddToolbox(r"D:/Program Files (x86)/ArcGIS/Desktop10.0/ArcToolbox/Toolboxes/Data Management Tools.tbx")
#arcpy.AddToolbox(r"D:/Program Files (x86)/ArcGIS/Desktop10.0/ArcToolbox/Toolboxes/Analysis Tools.tbx")


start_time = time.time()
today = date.today()


#ofoWS = r'L:\GISGroup\OFO_data\neteffarea_data'
ofoWS = r'L:\GISRestricted\OFO_data\neteffarea_data'


strftime("%Y-%m-%d %H:%M:%S")
print strftime("%Y-%m-%d %H:%M:%S") + "\n"

# Test if bastsp.shp Exists
if not arcpy.Exists(arcpy.Workspace + "\\" + "bastsp.shp"):
    print arcpy.Workspace + "\\" + "bastsp.shp does not exist - exiting script"


# Local variables...
#bastsp = arcpy.Workspace + "\\" + "bastsp.shp"
bastsp = arcpy.Workspace + "\\" + "bastsp.shp"
bastsp_Layer = arcpy.Workspace + "\\" + "bastsp_Layer"
test_Layer = arcpy.Workspace + "\\" + "xx1._Layer"
#neteffarea = arcpy.Workspace + "\\" + "neteffarea.shp"
neteffarea = arcpy.Workspace + "\\" + "fsa.gdb\\neteffarea"
test_Layer = arcpy.Workspace + "\\" + "test_Layer"
test_Layer = arcpy.Workspace + "\\" + "in_memory\\test_Layer"
neteffarea_Frequency = arcpy.Workspace + "\\" + "neteffarea_Frequency.dbf"
themonth = sys.argv[3]
#themonth = 'MAY'
 c = neteffarea.split('\\')
 if themonth.lower() == 'dec':
     currentneteff = 'nea_%s%s' % (themonth.lower(), today.year - 1)
 else:
     currentneteff = 'nea_%s%s' % (themonth.lower(), today.year)


 arcpy.AddMessage('Current Net Effective area dataset = ' + currentneteff)

 count = 0
 ribConn = r"l:\gisgroup\rib.gdb"
 luse = os.path.join(ribConn, "landuse_t")



 if arcpy.Exists(arcpy.Workspace + "\\" + "neteffarea_Frequency.dbf"):
     arcpy.Delete_management(arcpy.Workspace + "\\" +"neteffarea_Frequency.dbf")


 if arcpy.Exists(neteffarea):
     arcpy.Delete_management(neteffarea, "FeatureClass")

 if arcpy.Exists(neteffarea):
     arcpy.Delete_management(neteffarea, "FeatureClass")


 if arcpy.Exists(arcpy.Workspace + '\\' + currentneteff):
     arcpy.Delete_management(arcpy.Workspace + '\\' + currentneteff, "FeatureClass")


 ofoFC = os.path.join(ofoWS, currentneteff + '.shp')
 if arcpy.Exists(ofoFC):
```

```
#Process: Add Fields...
listFields = []
listFields.append('grpspec;TEXT;40')
listFields.append('AVAILABILITY;TEXT;40')
listFields.append('CAPABILITY;TEXT;40')
listFields.append('STANDING;TEXT;40')
listFields.append('CF;TEXT;40')
listFields.append('NOTPLANTED;TEXT;40')
listFields.append('OWNER;TEXT;40')
listFields.append('PLNOWNER;TEXT;40')
listFields.append('LANDOWNER;TEXT;40')
listFields.append('NETEFF_AREA;TEXT;40')
listFields.append('NETCPT_AREA;TEXT;40')
listFields.append('FORNAME;TEXT;40')
listFields.append('LANDESCRIP;TEXT;40')

print
arcpy.AddMessage(" ")

stepMsg = "2 Adding Fields"
arcpy.AddMessage(stepMsg)
print stepMsg

for i in listFields:
    x = i.split(';')
    arcpy.AddMessage("  %s" % x[0])
    arcpy.AddField_management(test_Layer, x[0],x[1],"","",x[2],"", "NULLABLE", "NON_REQUIRED", "")



print
arcpy.AddMessage(" ")

stepMsg = "3. Calculating Landuse"
arcpy.AddMessage(stepMsg)
print stepMsg

arcpy.AddJoin_management(test_Layer, "LANDUSE", luse, "LANDUSE", "KEEP_ALL")

fieldList = arcpy.ListFields(test_Layer)
for field in fieldList:
    print field.name

#print "Calculate field landescrip = landuse_t.descriptio"
arcpy.CalculateField_management(test_Layer, "test.LANDESCRIP", "[RIB.SDE.LANDUSE_T.DESCRIPTION]", "VB", "")
#print "Removing join landuse_t"
arcpy.RemoveJoin_management(test_Layer, "RIB.SDE.LANDUSE_T")

#print "Calculate field landescrip = landuse_t.descriptio"
arcpy.CalculateField_management(test_Layer, "LANDESCRIP", "[landuse_t.DESCRIPTION]", "VB", "")
#print "Removing join landuse_t"
arcpy.RemoveJoin_management(test_Layer)
arcpy.AddJoin_management(test_Layer, "LANDUSE", luse, "LANDUSE", "KEEP_ALL")

fieldList = arcpy.ListFields(test_Layer)
for field in fieldList:
    print field.name

#print "Calculate field landescrip = landuse_t.descriptio"
arcpy.CalculateField_management(test_Layer, "test.LANDESCRIP", "[RIB.SDE.LANDUSE_T.DESCRIPTION]", "VB", "")
#print "Removing join landuse_t"
arcpy.RemoveJoin_management(test_Layer, "RIB.SDE.LANDUSE_T")

#print "Calculate field landescrip = landuse_t.descriptio"
arcpy.CalculateField_management(test_Layer, "LANDESCRIP", "[landuse_t.DESCRIPTION]", "VB", "")
#print "Removing join landuse_t"
arcpy.RemoveJoin_management(test_Layer)

# calculate availabily to x
arcpy.CalculateField_management(test_Layer, "AVAILABILITY", "\"x\"", "VB", "")


# calculate species by group
print
arcpy.AddMessage(" ")
```

```
arcpy.SelectLayerByAttribute_management(test_Layer, "NEW_SELECTION", "\"SPECIES\" >= 5 AND \"SPECIES\" <=8")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"SPECIES\" >= 10 AND \"SPECIES\" <=23")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"SPECIES\" >= 25 AND \"SPECIES\" <=72")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"SPECIES\" >= 75 AND \"SPECIES\" <=76")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"SPECIES\" =79")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"SPECIES\" >= 99 AND \"SPECIES\" <=103")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"SPECIES\" >= 105 AND \"SPECIES\" <=125")
count = arcpy.GetCount_management(test_Layer)
print "  HARDWOOD - selected records = " + str(count)
arcpy.AddMessage("  HARDWOOD - selected records = " + str(count))
arcpy.CalculateField_management(test_Layer, "grpspec", "\"HARDWOOD\"", "VB", "")
arcpy.SelectLayerByAttribute_management(test_Layer, "CLEAR_SELECTION", "")

# calculate net effective area
print
arcpy.AddMessage(" ")
#spdescQry = "\"SP_DESC\" = 1"
spdescQry = "\"SP_DESC\" <= 2"

qryMsg = "Using query: %s" % spdescQry
if spdescQry[-1] == "2":
    qryMsg += "    includes SCATTERED TREES (orig)"
else:
    qryMsg +=  "    excludes SCATTERED TREES (upd)"

stepMsg = "5 Calculating NETEFF_AREA & AVAILABILITY"
arcpy.AddMessage(stepMsg)
print stepMsg


#arcpy.SelectLayerByAttribute_management(test_Layer, "NEW_SELECTION", "\"SP_DESC\" = 1 ")           #upd - excl scattered trees
#arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"EX_DESC\" = 1 ")
#arcpy.SelectLayerByAttribute_management(test_Layer, "NEW_SELECTION", "\"SP_DESC\" <= 2 ")           #orig - incl scattered trees
arcpy.SelectLayerByAttribute_management(test_Layer, "NEW_SELECTION", spdescQry)
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"EX_DESC\" >= 1 AND \"EX_DESC\" <=2")

#ds4 23.1.17 updated FIREBREAK QRY TO TAKE ACCOUNT OF 2016 Firebreak refinement (ie 1 = Internal, 2 = External)
#arcpy.SelectLayerByAttribute_management(test_Layer, "SUBSET_SELECTION", "\"LANDUSE\" >=1800 AND \"LANDUSE\" <=2999 AND \"FIREBREAK\" <> 1 ")
arcpy.SelectLayerByAttribute_management(test_Layer, "SUBSET_SELECTION", "\"LANDUSE\" >=1800 AND \"LANDUSE\" <=2999 AND \"FIREBREAK\" = 0 ")
arcpy.SelectLayerByAttribute_management(test_Layer, "SUBSET_SELECTION", "\"SPECIES\" >=1")
count = arcpy.GetCount_management(test_Layer)
print "  STANDING - selected records = " + str(count)
arcpy.AddMessage("  STANDING - selected records = " + str(count))
print qryMsg
arcpy.AddWarning(qryMsg)
arcpy.CalculateField_management(test_Layer, "STANDING", "\"STANDING\"", "VB", "")
arcpy.CalculateField_management(test_Layer, "NETEFF_AREA", "\"NETEFF\"", "VB", "")
arcpy.CalculateField_management(test_Layer, "AVAILABILITY", "\"STANDING\"", "VB", "")
arcpy.SelectLayerByAttribute_management(test_Layer, "CLEAR_SELECTION", "")

# calculate cf
arcpy.SelectLayerByAttribute_management(test_Layer, "NEW_SELECTION", "\"ST_OPN\" = 'CT' ")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"ST_OPN\" = 'CU'")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"ST_OPN\" = 'BR'")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"ST_OPN\" = 'LS'")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"ST_OPN\" = 'FS'")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"ST_OPN\" = 'ZV'")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"ST_OPN\" = 'ZE'")

#ds4 23.1.17 updated FIREBREAK QRY TO TAKE ACCOUNT OF 2016 Firebreak refinement (ie 1 = Internal, 2 = External)
#arcpy.SelectLayerByAttribute_management(test_Layer, "SUBSET_SELECTION", "\"LANDUSE\" >=1800 AND \"LANDUSE\" <=2999 AND \"FIREBREAK\" <> 1 ")
arcpy.SelectLayerByAttribute_management(test_Layer, "SUBSET_SELECTION", "\"LANDUSE\" >=1800 AND \"LANDUSE\" <=2999 AND \"FIREBREAK\" = 0 ")
arcpy.SelectLayerByAttribute_management(test_Layer, "SUBSET_SELECTION", "\"SP_SUIT\" >=1 AND \"SP_SUIT\" <=3")
count = arcpy.GetCount_management(test_Layer)
print "  CF - selected records = " + str(count)
arcpy.AddMessage("  CF - selected records = " + str(count))
arcpy.CalculateField_management(test_Layer, "CF", "\"CF\"", "VB", "")
arcpy.CalculateField_management(test_Layer, "NETEFF_AREA", "\"CF\"", "VB", "")
arcpy.CalculateField_management(test_Layer, "AVAILABILITY", "\"CF\"", "VB", "")
arcpy.SelectLayerByAttribute_management(test_Layer, "CLEAR_SELECTION", "")



# calculate areas not planted
arcpy.SelectLayerByAttribute_management(test_Layer, "NEW_SELECTION", "\"LANDUSE\" = 3010 ")
arcpy.SelectLayerByAttribute_management(test_Layer, "ADD_TO_SELECTION", "\"LANDUSE\" = 3020")
arcpy.SelectLayerByAttribute_management(test_Layer, "REMOVE_FROM_SELECTION", "\"SPECIES\" >= 1")
arcpy.SelectLayerByAttribute_management(test_Layer, "REMOVE_FROM_SELECTION", "\"NETEFF_AREA\" = 'CF'")
count = arcpy.GetCount_management(test_Layer)
print "  NOTPLANTED - selected records = " + str(count)
```


Net Effective Area (NEA)
Purpose & Context
The Net Effective Area (NEA) is a core GIS derived dataset that describes sub-compartment landuse across the GT Forests estate.
Input Data
1. Base - the base layer provides a coarse landuse description and land owner and plantation owner information for planted and non-planted areas. All other geometry should sit within the base geometry.
2. Stocking - the stocking layer provides a quarterly snapshot of all productive plantation areas and their previous operation type (e.g. T1, T2, CF etc.)
3. Site Productivity (sp) - the sp layer provides high resolution productivity index information for all productive plantation areas.
4. Experiments - the experiments layer provides information on all forest experiments that occur within productive plantation areas. Some experiments are not included in some reports.
Assumptions
This process assumes that all input data is prepared accurately prior to the production of the NEA. Some exception reporting is included, but errors in the input datasets will propagate into the NEA if not accounted for.
NEA Inputs
Currently, all core datasets are extracted from their underlying resources and placed in an FGDB to be used specifically for NEA generation.
Current Quarter Data
base
stk
sp
Experiments
dbo.L..._POLY
NEA Processing
Geometrical Unions
The main geometry transformer in the NEA generation is a union of the 4 core datasets.
AreaOnAreaOverlayer
Area
Area
Remnants
<Rejected>
All of the core data should sit within the base, anything that doesn't is an exception.
Tester
Passed
Failed
Exclude Clarkes non-PLA
The area at Clarkes is excluded because it is not in the PLA but is in the base (see Resources to understand why!)
Tester_3
Passed
Failed
Reclassifying Attributes
All of the NEA attributes are derived from some combination of the input data fields. There is complex logic that determines each attribute's value that can be viewed inside the transformer.
AttributeManager
Output
NEA Outputs
The Current NEA is overwritten in the GIS Reference EGDB.
AttributeManager_20
Output
FME_Current_NEA
The NEA is also produced as a shp and placed in a folder structure in the GIS Restricted folder.
NEA_FME

OneFortyOne

We manage over 160,000 ha of land for softwood plantations and conservation.

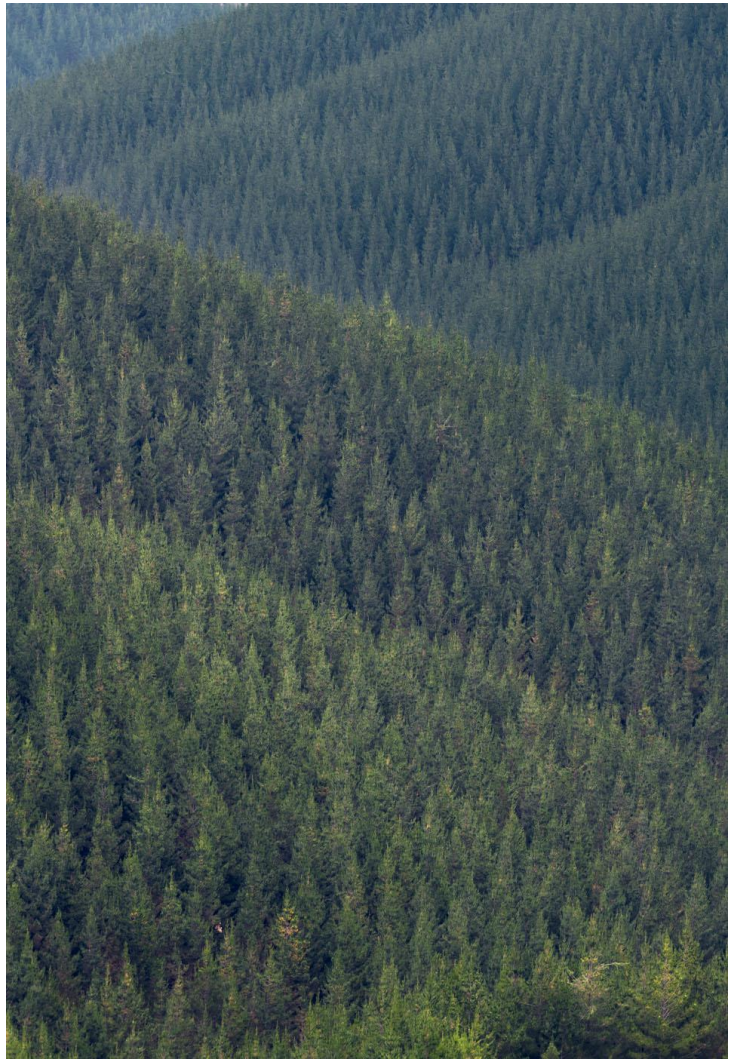
We are a forestry and sawmill business.

We process timber at two sawmills in Mount Gambier and Kaituna.

We operate throughout the Green Triangle in Australia and Nelson region in New Zealand.

Green Triangle Forest Australia

Nelson Forest
New Zealand



Our GIS

Reporting

We use ESRI dashboards and web apps as well as more common tools like Excel and Power BI for reporting.

Applications

We use Trimble's Land Resource Manager for core operational planning.

We use ESRI apps like Survey123 and Field Maps for mobile work.

We use FME for managing complex data workflows.

ArcGIS Enterprise

We have an ArcGIS Enterprise setup with Portal for web maps and apps, and ArcGIS Pro for desktop analysis.

Core Infrastructure

Our corporate data is stored in SQL Server Enterprise Geodatabases.

Outline

Core GIS Workflows

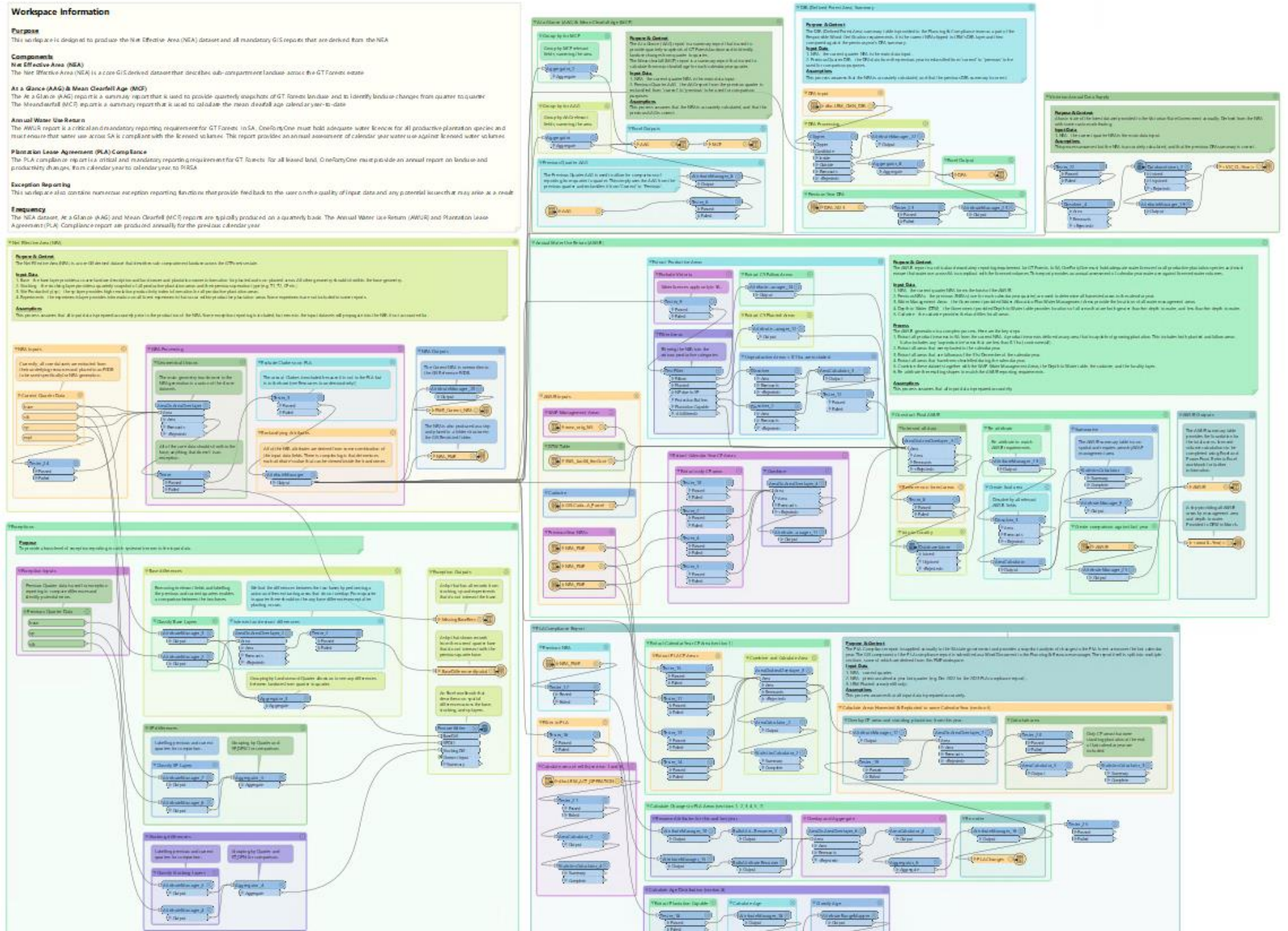
Overhauling critical Python-based GIS workflows into a 90 sec FME workspace.

New Horizons

Automating new GIS solutions, including pulling lightning strike data into web maps to assist in forest fire management.

Enhancing Existing Tools with FME

Combining Python and FME to make existing workflows usable by non-experts.



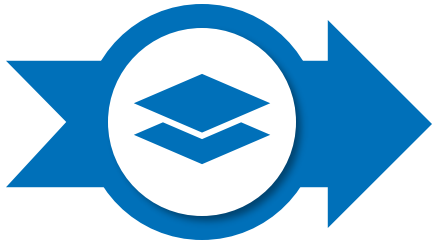
Core GIS Workflows

Core GIS Workflows

Preparation

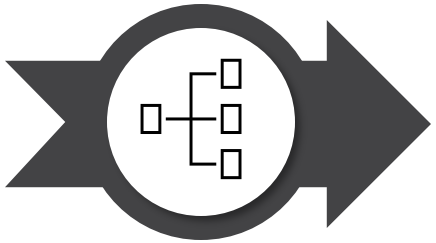
Core input data needs to be prepared.

Includes basic data integrity and topology checks.



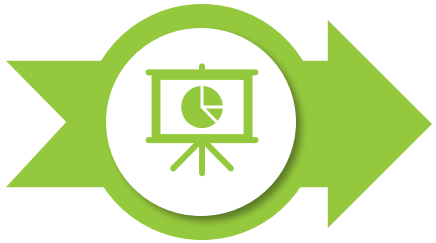
Intersection

Core input datasets are spatially intersected to create composite dataset.



Attribute Mapping

New reporting attributes are mapped based on combinations of core data attributes.

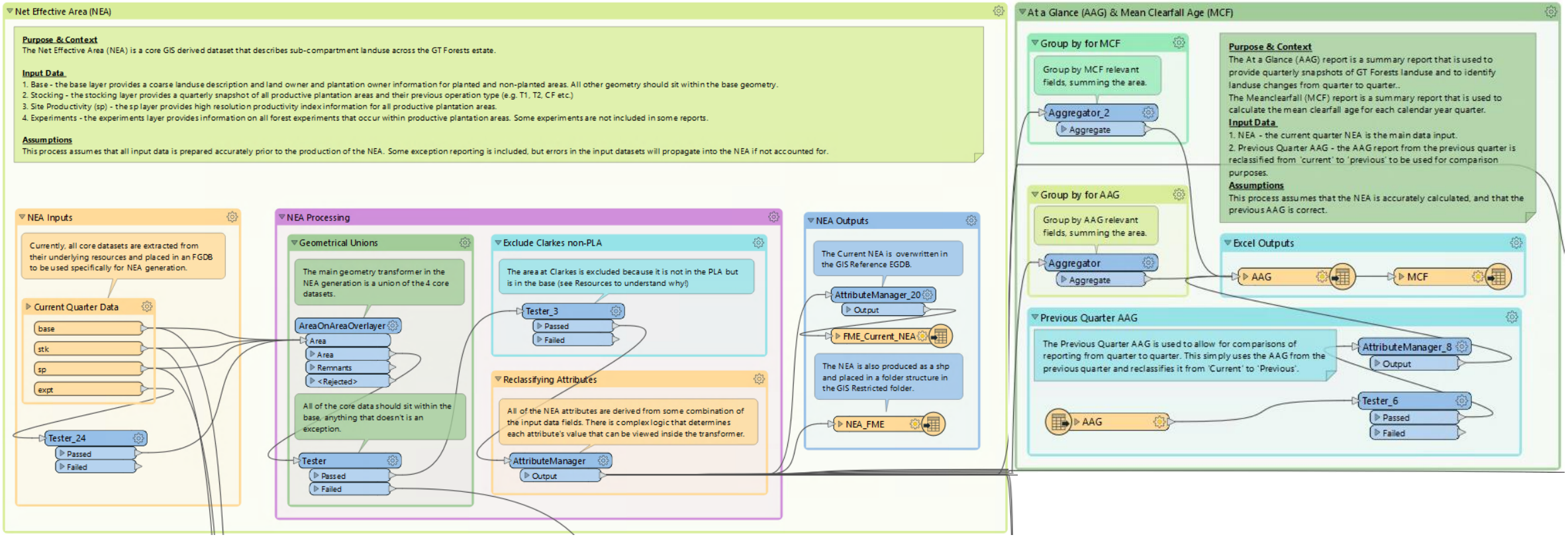


Reporting

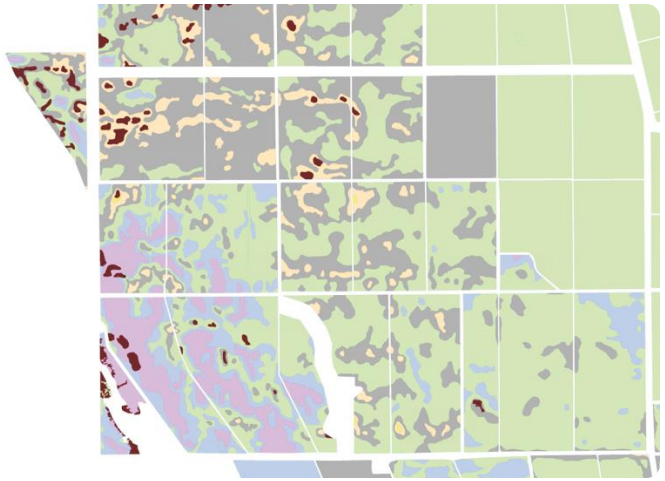
Specific reports are generated in external tools from single source composite dataset.

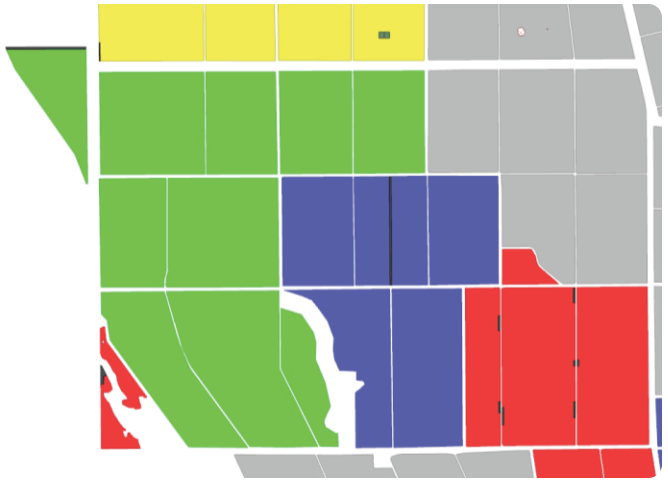
Some post-processing required but all completed in FME.

Core GIS Workflow

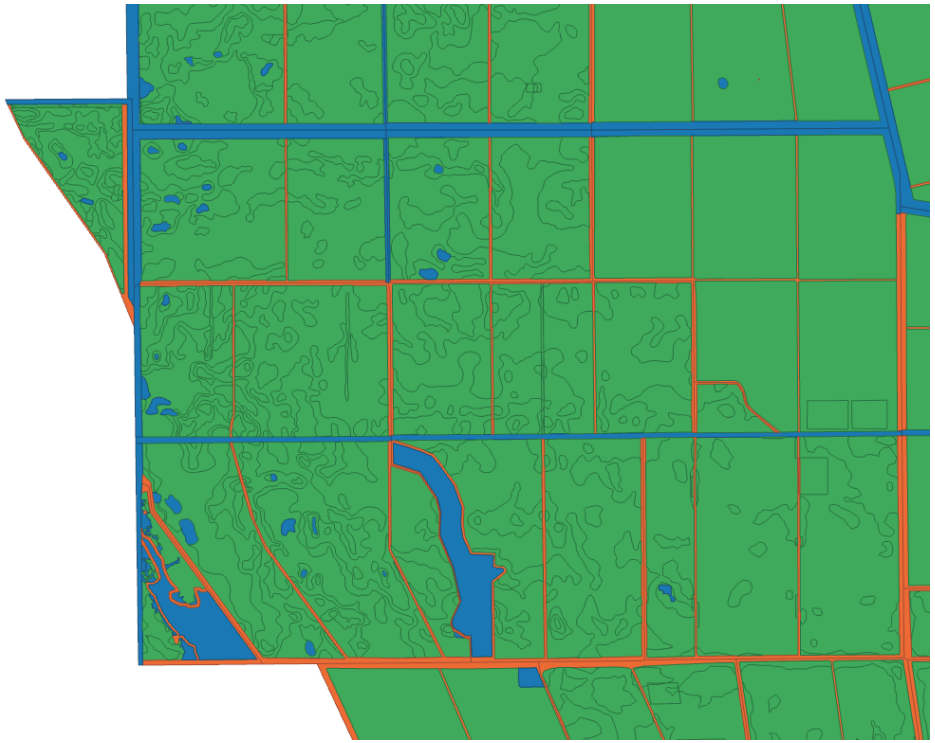


Preparation



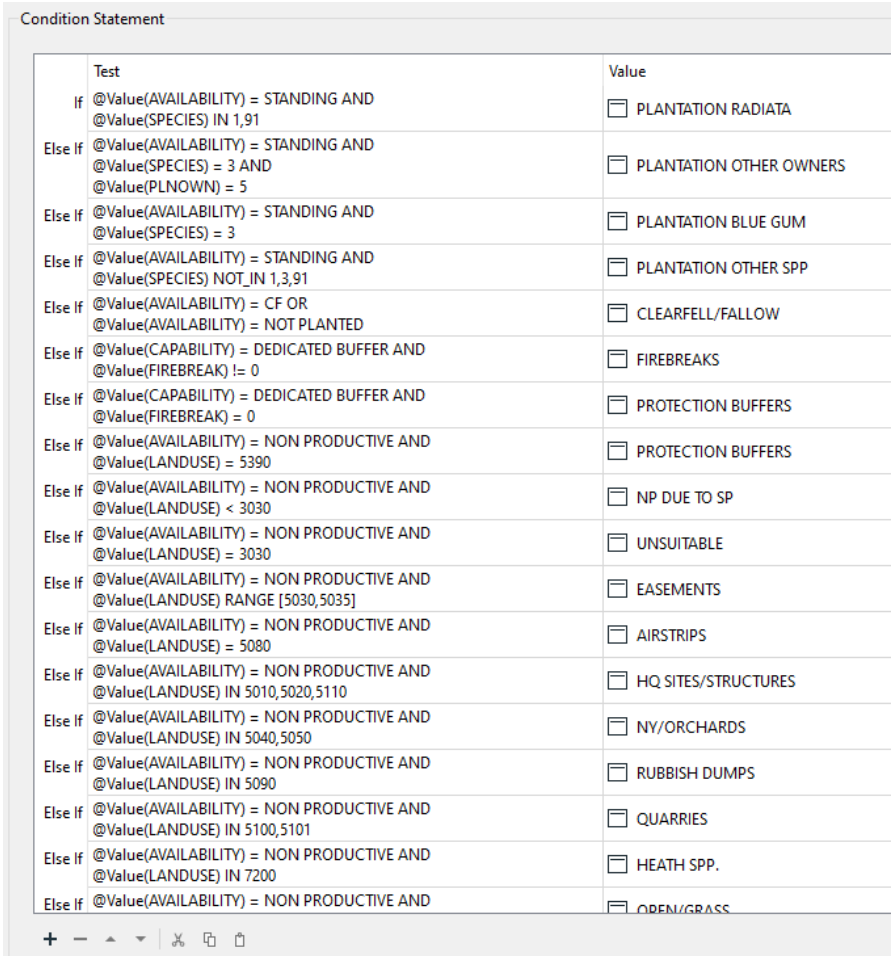


Intersection



Attribute Mapping

Area_HA	@Area()/10000	real64
grpspec	6 Possible Values	varchar(200)
NETEFF_AREA	4 Possible Values	varchar(200)
AVAILABILITY	4 Possible Values	varchar(200)
CAPABILITY	4 Possible Values	varchar(200)
STANDING	2 Possible Values	varchar(200)
CF	2 Possible Values	varchar(200)
NOTPLANTED	2 Possible Values	varchar(200)
PLNOWNER	4 Possible Values	varchar(200)
LANDOWNER	6 Possible Values	varchar(200)
NETCPT_AREA	2 Possible Values	varchar(200)
FORNAME	8 Possible Values	varchar(200)
AAG_CODE	29 Possible Values	varchar(200)
CATEGORY	8 Possible Values	varchar(200)
AGE	3 Possible Values	int16
AAG_CODE_GROUP	4 Possible Values	varchar(200)
CATEGORY_GROUP	3 Possible Values	varchar(200)
Quarter	Current	buffer



Reporting

Sum of Area_HA	Column Labels	
	Current	Previous
Row Labels		
TOTAL PLANTATION CAPABLE	**82,466.48**	**82,333.08**
PLANTATION RADIATA	80,194.98	80,439.43
CLEARFELL/FALLOW	1,857.85	1,480.01
PLANTATION BLUE GUM	22.09	22.09
PLANTATION OTHER SPP	81.29	81.29
PLANTATION OTHER OWNERS	310.27	310.27
TOTAL PROTECTION BUFFERS	**8,134.71**	**8,134.75**
FIREBREAKS	7,679.18	7,679.18
PROTECTION BUFFERS	455.52	455.56
TOTAL NON-PRODUCTIVE AREAS	**5,549.63**	**5,682.99**
AIRSTRIPS	86.09	86.09
BIODIVERSITY CORRIDORS	92.91	92.91
CULTURAL SITES	8.30	8.30
DAMS	15.13	15.13
EASEMENTS	636.77	636.76
FAUNA CONSERVATION	0.21	0.21
HEATH SPP.	64.47	64.47
HISTORIC SITES	6.75	6.75
HQ SITES/STRUCTURES	81.09	81.09
NATIVE VEGETATION	1,674.95	1,674.95
NP DUE TO SP	930.47	1,063.83
NY/ORCHARDS	220.24	220.24
OPEN/GRASS	128.34	128.34
QUARRIES	131.40	131.40
RECREATION SITES	9.39	9.39
ROCKY/KARST	37.10	37.10
RUBBISH DUMPS	8.00	8.00
SWAMP VEGETATION	1,182.19	1,182.19
UNSUITABLE	202.50	202.50
WATERCOURSES	33.35	33.35
Grand Total	**96,150.82**	**96,150.82**

Values are shown as difference from previous year (i.e. 2024 is the 2024 value - the 2023 value). Decrease in Water Use. Increase in Water Use.

MANAGEMENT	Licence #	Forest Type	Values Total Area (HA) 2023	Values Total Area (HA) 2024	Variance (ML) 2023	Variance (ML) 2024	Variance (ML) 2024	Total Fallow (HA) 2023	Total Fallow (HA) 2024	< 6m DTW (HA) 2023	< 6m DTW (HA) 2024	< 6m DTW Fallow (HA) 2023	< 6m DTW Fallow (HA) 2024	Recharge (ML) 2023	Recharge (ML) 2024
BENARA	241477				-0.01	637.36	644.11		6.75		2.35		-0.01		2.05		-3.33
BLANCHE CENTRAL	241478				-0.01	439.00	440.70		1.70		1.16						-1.70
BOWAKA	241479					13.22	13.22										
BRAY	241481				0.01	145.45	148.13		2.68		3.58						-2.68
COLES	241482					12.65	14.33		1.68		0.63				0.63		-0.63
COMAUM	241485					242.42	242.27		-0.14		-0.29						0.14
COMPTON	241486				-0.01	197.22	200.94		3.71		2.55						-3.71
DONOVANS	241487				0.10	899.78	910.35		10.57		7.39						-10.57
FOX	241488					113.16	113.16										
GLENBURNIE	241489				-0.01	1,702.19	1,708.71		6.53		5.21						-6.53
GREY	241490				0.00	41.49	41.62		0.12		0.10						-0.12
HINDMARSH	241491				-3.74	2,669.19	2,685.73		16.54		4.13		0.40		4.44		-9.84
JOANNA	241492				0.00	166.60	166.86		0.26		0.18				0.11		-0.08
KENNION	241493				-33.06	693.99	700.15		6.16		-29.66		-33.28		-31.62		-3.40
KONGORONG	241494					94.85	106.83		11.99		8.50						-11.99
MACDONNELL	305749					18.50	18.50										
MONBULLA	241495					153.52	286.42		132.90		42.19				42.19		-62.86
MOORAK	241496					9.24	9.24										
MOUNT BENSON	241497					577.86	579.34		1.48		2.12				0.25		-1.06
MOUNT MUIRHEAD	241498				-0.01	490.26	499.42		9.16		4.15		-0.01		3.23		-3.79
MYORA	241499				-0.00	1,219.80	1,226.74		6.94		5.22						-6.94
RIDDOCH	241500				0.24	1,593.95	1,549.32		-44.63		-41.16		-0.01		0.04		44.71
SHORT	241501				-0.01	293.66	295.67		2.01		0.68		-0.01		0.68		-0.86
SMITH	241502				-0.01	97.90	98.80		0.90		0.35		-0.01		0.35		-0.30
WATERHOUSE	241503					61.26	61.26										
YOUNG	241504				-3.90	2,017.52	1,991.13		-26.39		-11.51		-3.90		-12.19		12.63
ZONE 2A	241505				-6.43	3,465.86	3,434.39		-31.47		-27.05		-8.89		-13.44		23.92
ZONE 3A	241506				0.49	110.69	106.21		-4.48		-3.77		-0.01		-0.14		4.26
Grand Total					-46.36	18,178.60	18,293.54		114.94		-22.95		-45.73		-3.42		-44.71

Advantages of FME

01

Repeatability

Executed routinely with refreshed data, always with consistent and reliable results, and in < 2 minutes.

02

Visibility

Very easy to see what data processing is occurring and how attributes are mapped when clarification is needed.

03

Testability

Quick to determine where errors are occurring, and error checking can be done before processing.

04

Documentability

If documented correctly, the workspace becomes the metadata.

05

Accessibility

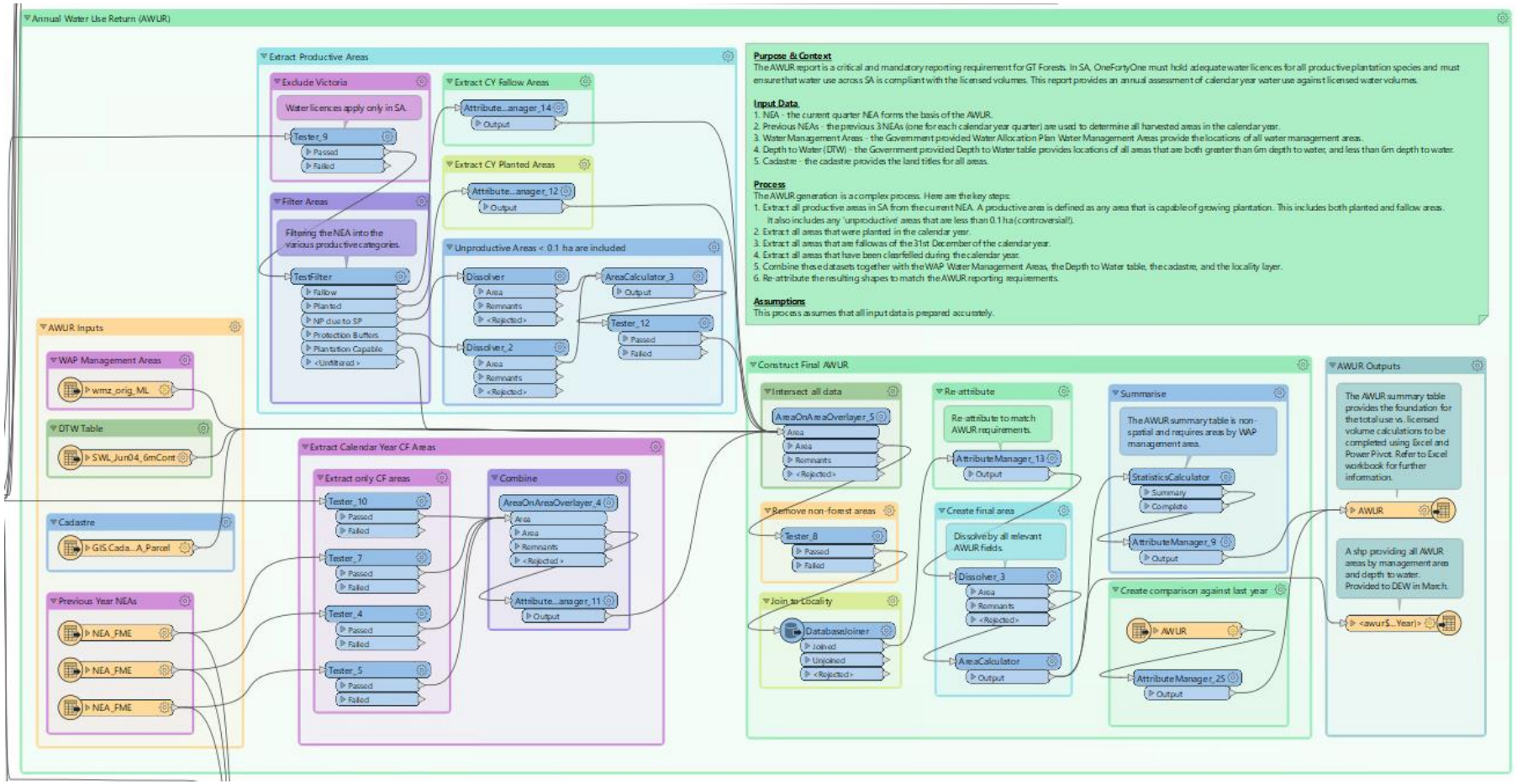
Non-expert users can easily visualize data flows and create their own – no programming experienced needed.

06

Extensibility

Very simple to add new functionality, just add a new bookmark and pick your transformers!

Extensions

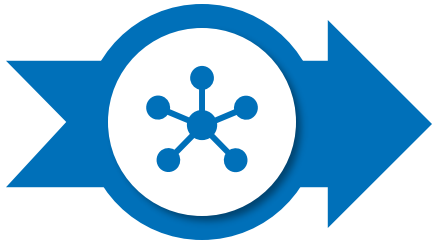

Annual Water Use Return (AWUR)
Extract Productive Areas
Exclude Victoria
Water licences apply only in SA.
Tester_9
Passed
Failed
Extract CY Fallow Areas
Attribute...anager_14
Output
Extract CY Planted Areas
Attribute...anager_12
Output
Filter Areas
Filtering the NEA into the various productive categories.
TestFilter
Fallow
Planted
NP due to SP
Protection Buffers
Plantation Capable
<Unfiltered>
Unproductive Areas < 0.1 ha are included
Dissolver
Area
Remnants
<Rejected>
AreaCalculator_3
Output
Dissolver_2
Area
Remnants
<Rejected>
Tester_12
Passed
Failed
AWUR Inputs
WAP Management Areas
wmz_orig_ML
DTW Table
SWL_Jun04_6mCont
Cadastre
GIS.Cada...A_Parcel
Previous Year NEAs
NEA_FME
NEA_FME
NEA_FME
Extract Calendar Year CF Areas
Extract only CF areas
Tester_10
Passed
Failed
Tester_7
Passed
Failed
Tester_4
Passed
Failed
Tester_5
Passed
Failed
Combine
AreaOnAreaOverlayer_4
Area
Area
Remnants
<Rejected>
Attribute...anager_11
Output
Purpose & Context
The AWUR report is a critical and mandatory reporting requirement for GT Forests. In SA, OneFortyOne must hold adequate water licences for all productive plantation species and must ensure that water use across SA is compliant with the licensed volumes. This report provides an annual assessment of calendar year water use against licensed water volumes.
Input Data
1. NEA - the current quarter NEA forms the basis of the AWUR.
2. Previous NEAs - the previous 3 NEAs (one for each calendar year quarter) are used to determine all harvested areas in the calendar year.
3. Water Management Areas - the Government provided Water Allocation Plan Water Management Areas provide the locations of all water management areas.
4. Depth to Water (DTW) - the Government provided Depth to Water table provides locations of all areas that are both greater than 6m depth to water, and less than 6m depth to water.
5. Cadastre - the cadastre provides the land titles for all areas.
Process
The AWUR generation is a complex process. Here are the key steps:
1. Extract all productive areas in SA from the current NEA. A productive area is defined as any area that is capable of growing plantation. This includes both planted and fallow areas. It also includes any 'unproductive' areas that are less than 0.1 ha (controversial!).
2. Extract all areas that were planted in the calendar year.
3. Extract all areas that are fallow as of the 31st December of the calendar year.
4. Extract all areas that have been clearfelled during the calendar year.
5. Combine these datasets together with the WAP Water Management Areas, the Depth to Water table, the cadastre, and the locality layer.
6. Re-attribute the resulting shapes to match the AWUR reporting requirements.
Assumptions
This process assumes that all input data is prepared accurately.
Construct Final AWUR
Intersect all data
AreaOnAreaOverlayer_5
Area
Area
Remnants
<Rejected>
Re-attribute
Re-attribute to match AWUR requirements.
AttributeManager_13
Output
Summarise
The AWUR summary table is non-spatial and requires areas by WAP management area.
StatisticsCalculator
Summary
Complete
AttributeManager_9
Output
Remove non-forest areas
Tester_8
Passed
Failed
Create final area
Dissolve by all relevant AWUR fields.
Dissolver_3
Area
Remnants
<Rejected>
AreaCalculator
Output
Join tor Locality
DatabaseJoiner
Joined
Unjoined
<Rejected>
Create comparison against last year
AWUR
AttributeManager_25
Output
AWUR Outputs
The AWUR summary table provides the foundation for the total use vs. licensed volume calculations to be completed using Excel and Power Pivot. Refer to Excel workbook for further information.
AWUR
A shp providing all AWUR areas by management area and depth to water. Provided to DEW in March.
<awur$...Year)>

Lightning Web Maps

Gather

Lightning data sits on an external web server owned by a 3rd party.

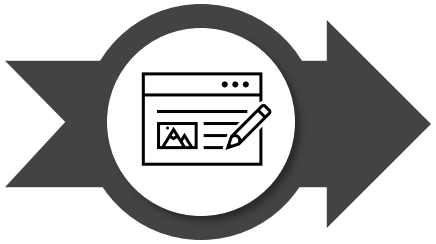
We can query but data is only exposed for a short time.



Collate

We systematically query the data and group it by time.

We adjust some attributes and align with our GIS.



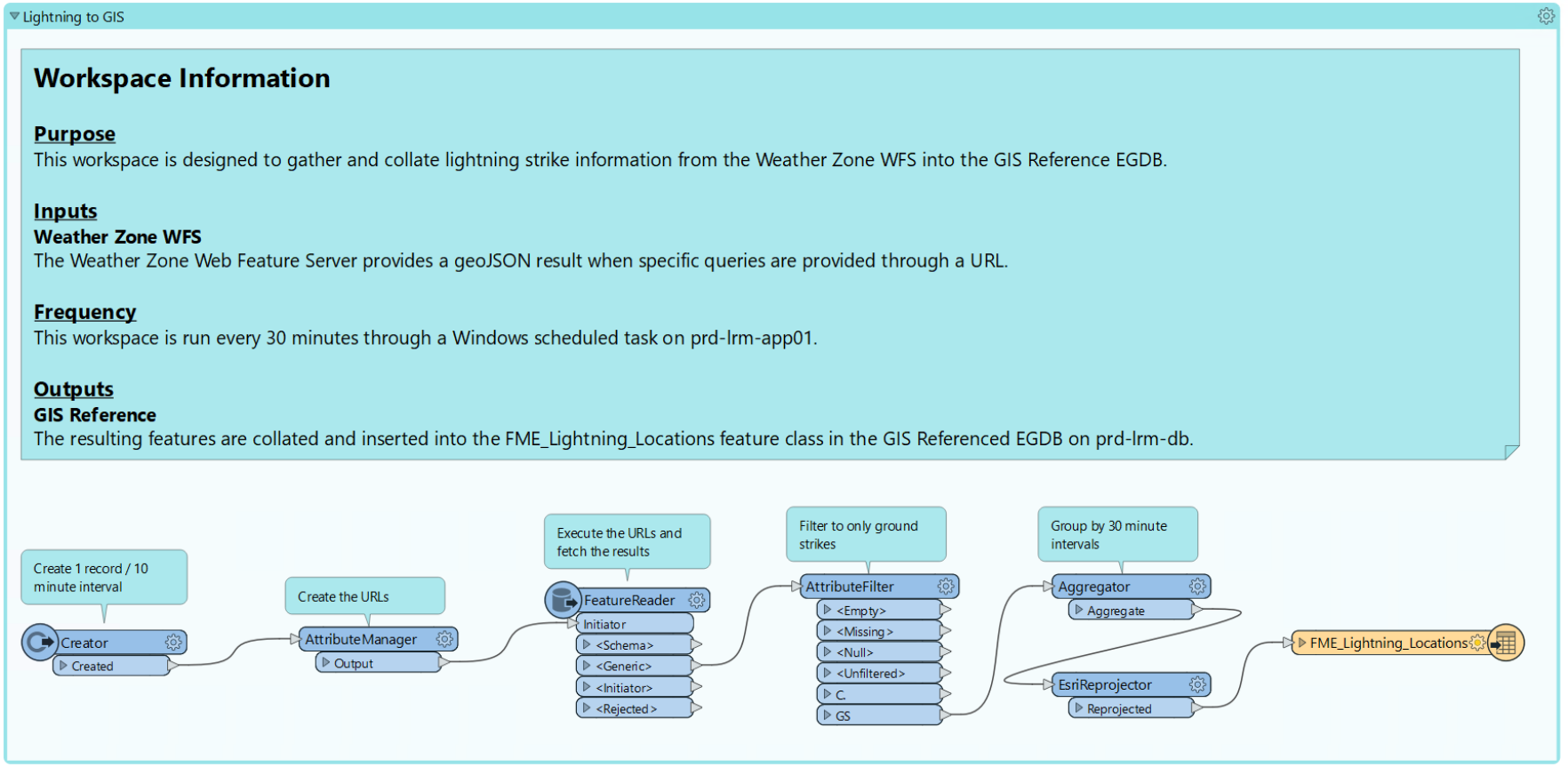
Publish

We insert the grouped and collated records into our Enterprise Geodatabase.

Data is now stored for historical events as well.

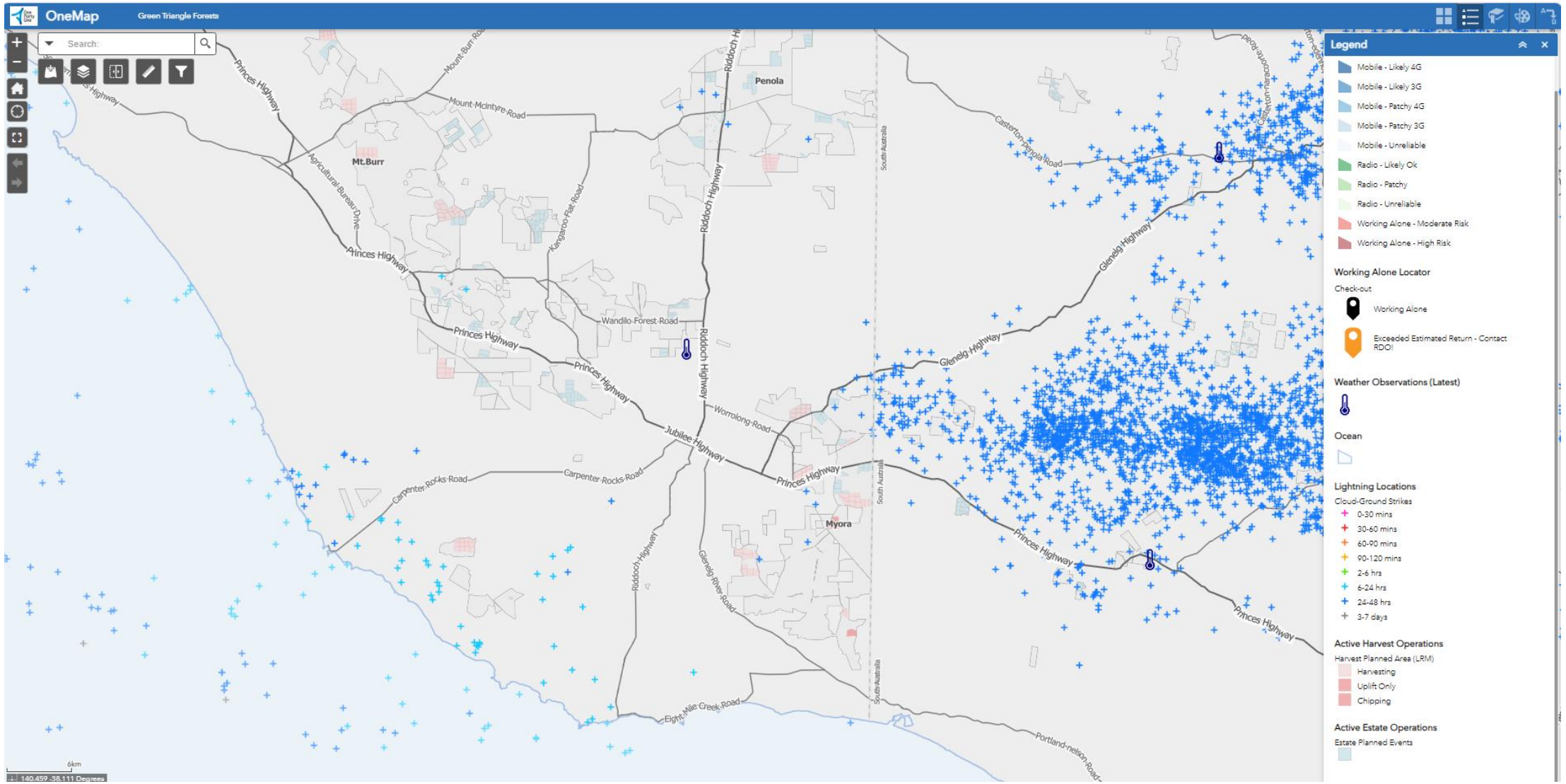
Consume

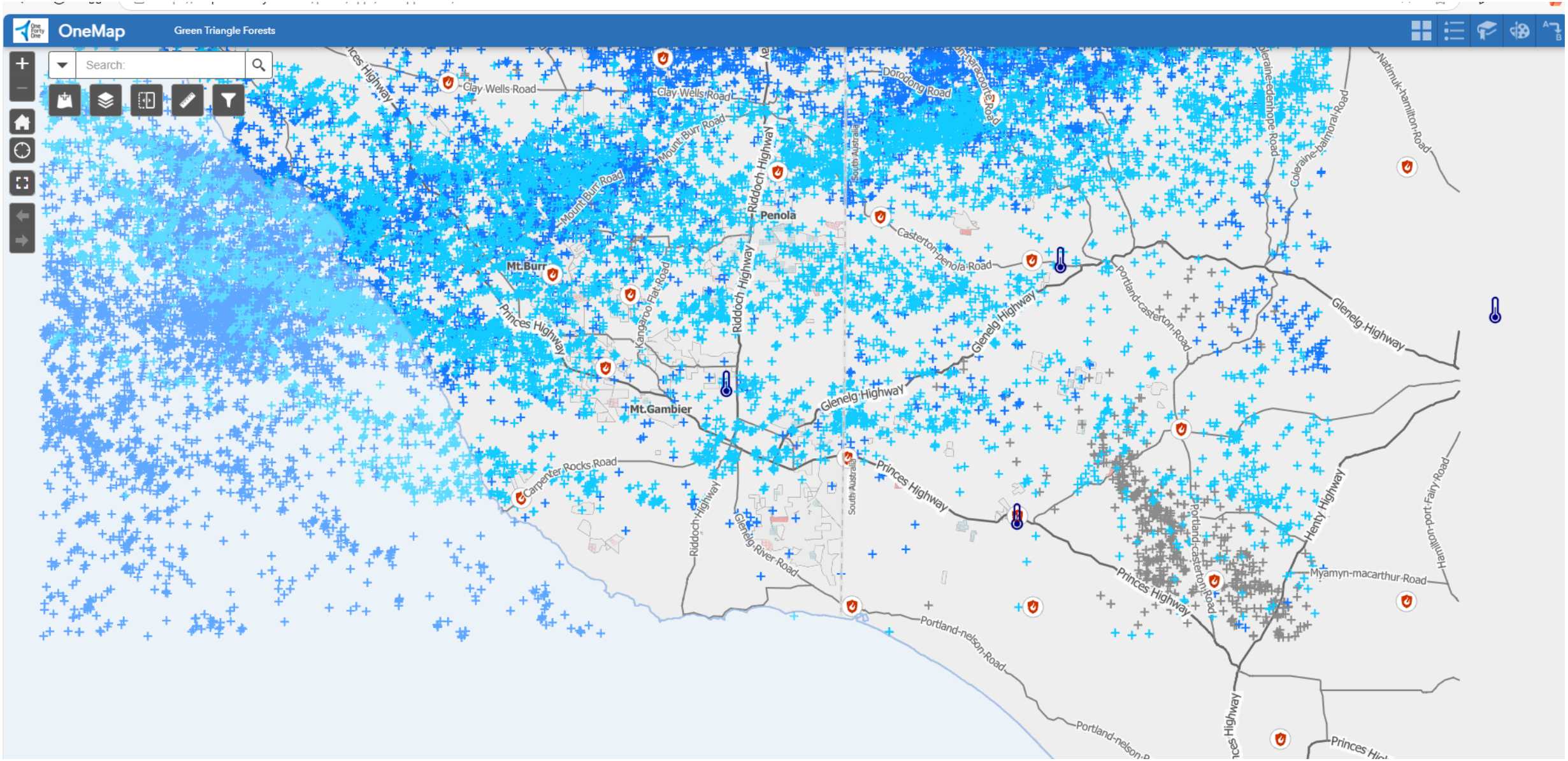
Our EGDB is synchronized with our ArcGIS Portal, so any inserts are instantly read into our web maps.



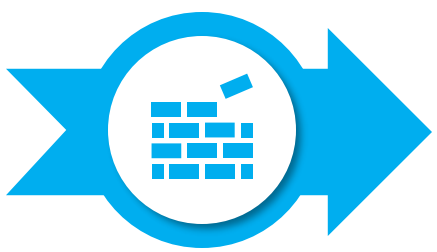
Attribute Actions

Input Attribute	Output Attribute	Value
_creation_instance	_creation_instance	<Enter value (optional)>
	TimeDuration	-PT@Evaluate(@Value(_creation_instance)*10)M
	TimeStamp	@TimeZoneRemove(... (MultiLine)
	TimeGroup	@DateTimeRound(@TimeZoneSet(@Value(TimeStamp),utc),down,minutes,30)
	bbox	139.627991,-38.082690,141.786804,-36.925743
	URL	https://geo.theweather.com.au/external-onefortyone/wfs?service=WFS&version=1.0.0&request=GetFeature&typeName=wztsli:lightning...
<Expose existing attribute>	<Add new attribute>	


OneMap
Green Triangle Forests
Search:
Legend
Mobile - Likely 4G
Mobile - Likely 3G
Mobile - Patchy 4G
Mobile - Patchy 3G
Mobile - Unreliable
Radio - Likely Ok
Radio - Patchy
Radio - Unreliable
Working Alone - Moderate Risk
Working Alone - High Risk
Working Alone Locator
Check-out
Working Alone
Exceeded Estimated Return - Contact RDO!
Weather Observations (Latest)
Ocean
Lightning Locations
Cloud-Ground Strikes
0-30 mins
30-60 mins
60-90 mins
90-120 mins
2-6 hrs
6-24 hrs
24-48 hrs
3-7 days
Active Harvest Operations
Harvest Planned Area (LRM)
Harvesting
Uplift Only
Chipping
Active Estate Operations
Estate Planned Events
Penola
Mt.Burr
Myora
Princes Highway
Riddoch Highway
Glenelg Highway
Jubilee Highway
Mount-Burr-Road
Mount-Mcintyre-Road
Agricultural Bureau Drive
Kangaroo-Flat-Road
Wandilo-Forest-Road
Worrolong-Road
Carpenter-Rocks-Road
Glenelg-River-Road
Eight-Mile-Creek-Road
Portland-nelson-Road
Casterton-Penola-Road
South Australia
6km
140.459 -38.111 Degrees


OneMap
Green Triangle Forests
Search:
Clay Wells Road
Mount Burr Road
Riddoch Highway
Penola
Casterton-penola Road
Mt.Burr
Princes Highway
Kangaroo Flat Road
Glenelg Highway
Portland-casterton Road
Mt.Gambier
Carpenter Rocks Road
Glenelg River Road
Portland-nelson Road
Henty Highway
Myamyn-macarthur Road
Hamilton-port Fairy Road
Natimuk-hamilton Road
Coleraine-balmoral Road
Dergholm-tarrington Road
Dododong Road
South Australia

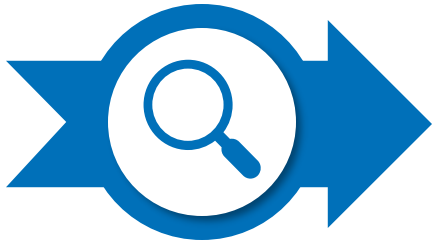
Enhancing Existing Tools



Deconstruct

Highly specialized or embedded existing workflows can be very difficult to understand, even for the workflow owners.

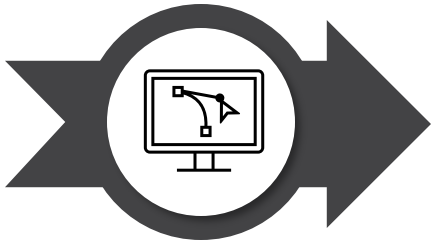
Breaking these workflows down into their key components is critical.



Identify

Identify which components are immutable and which can be improved or replaced.

Often you will find it's only a few key pieces that the workflow owner is attached to!



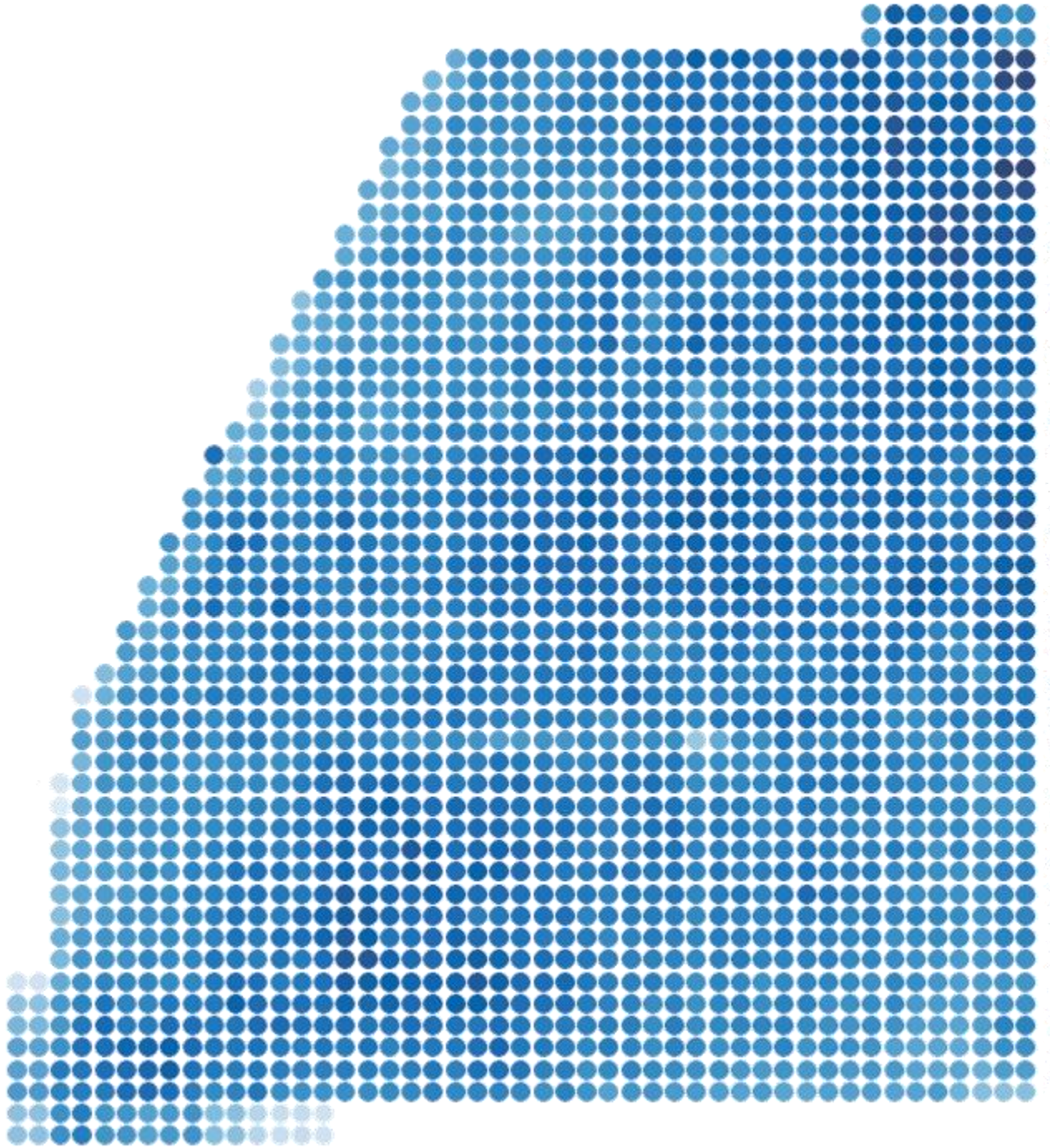
Enhance

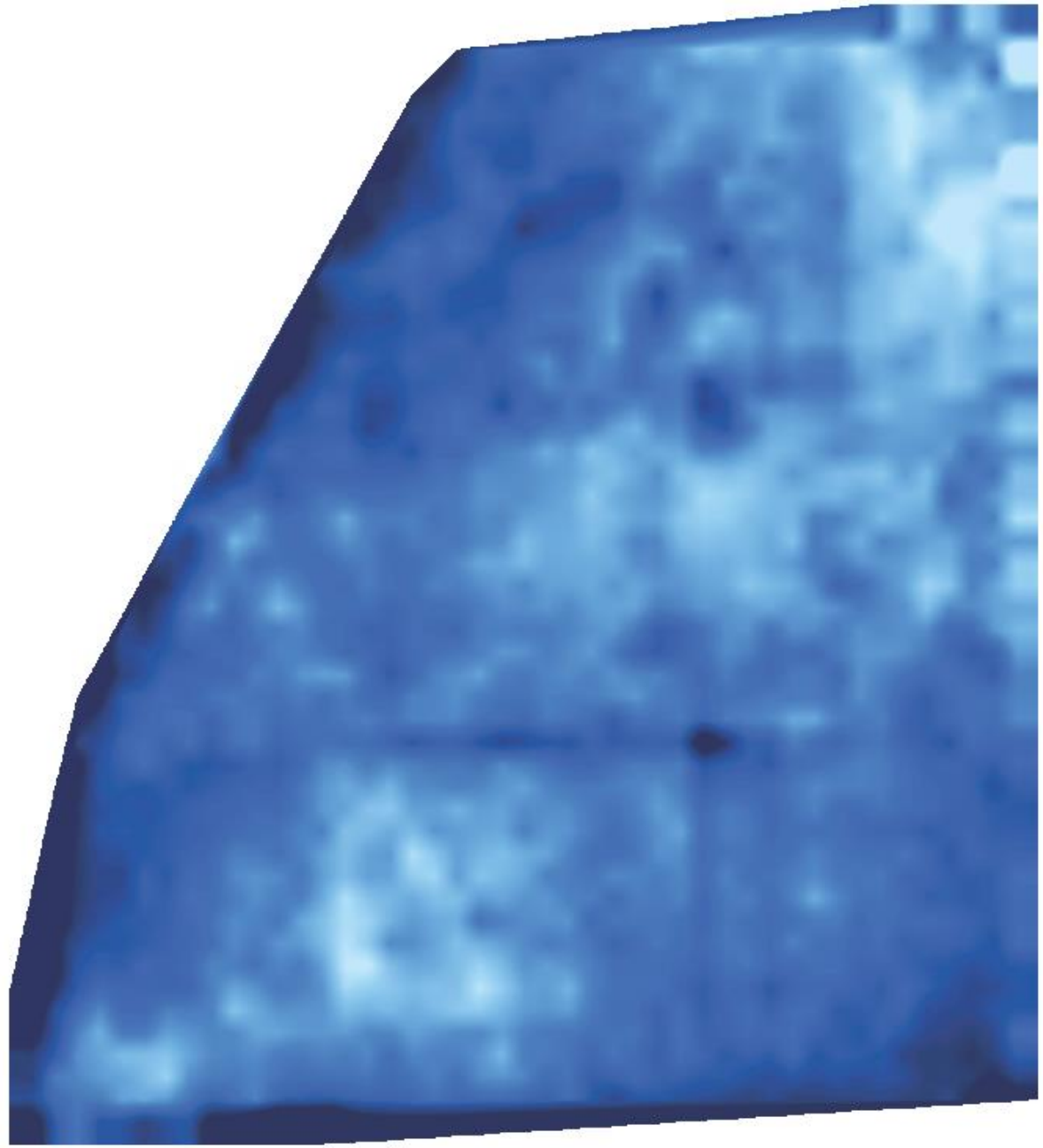
Replace the identified components with FME analogues and integrate the immutable components either in FME transformers or as readers.

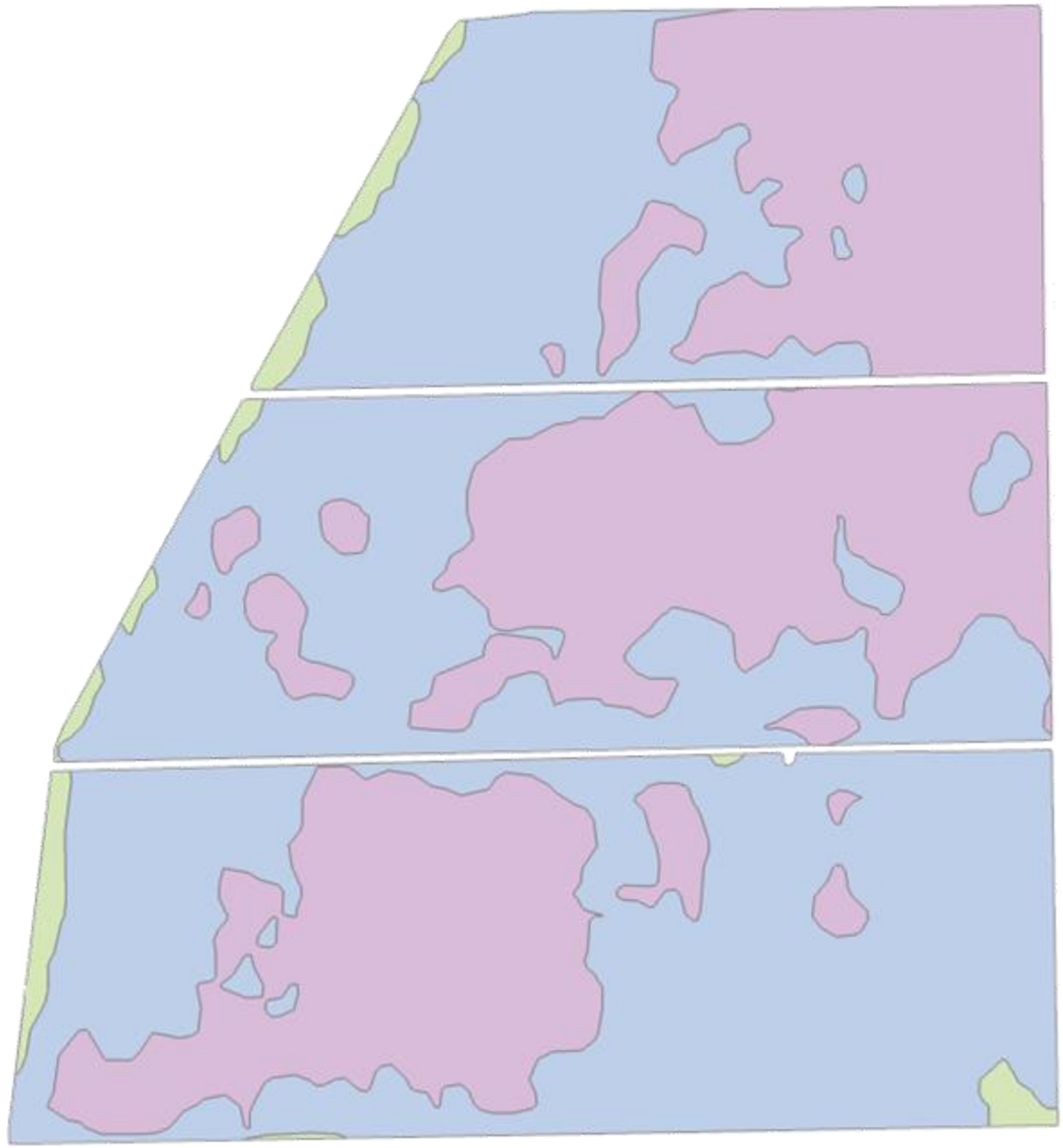
Educate

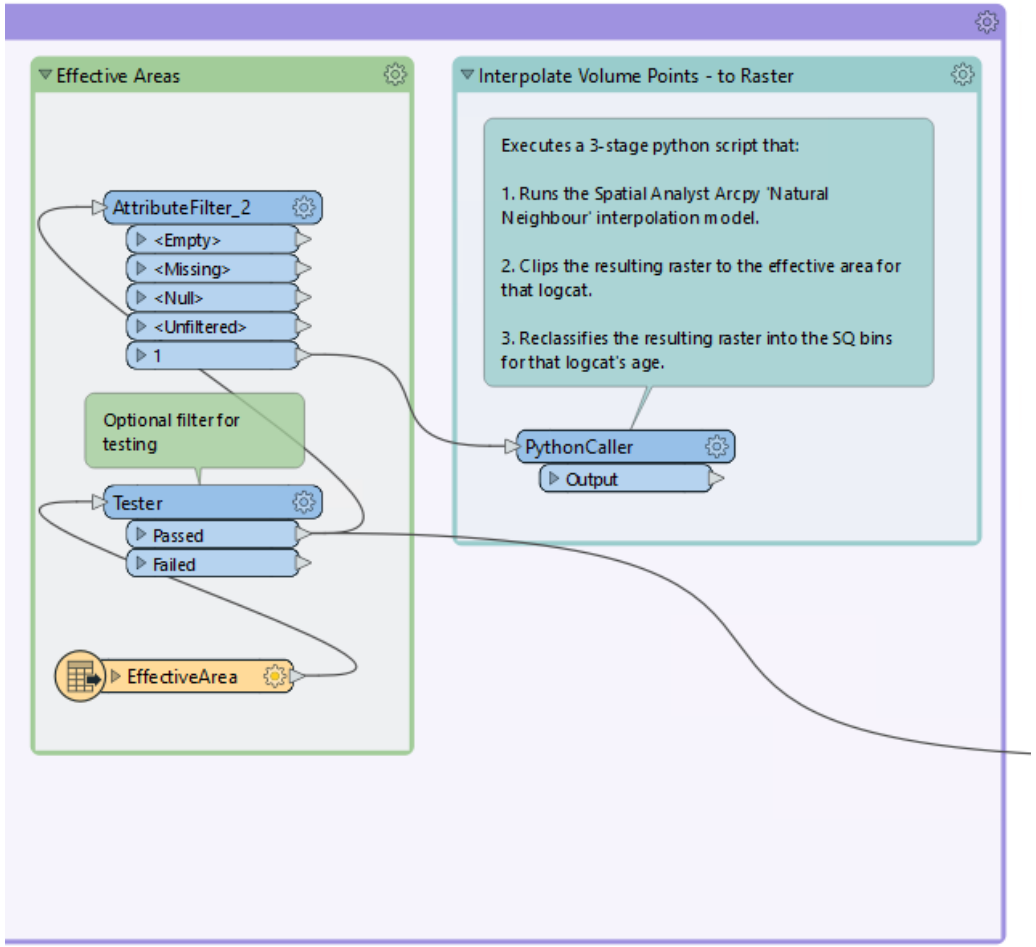
How did we let it get so bad?

It's rarely intentional – what can we change about our workflow approach to increase data sustainability across the organisation?

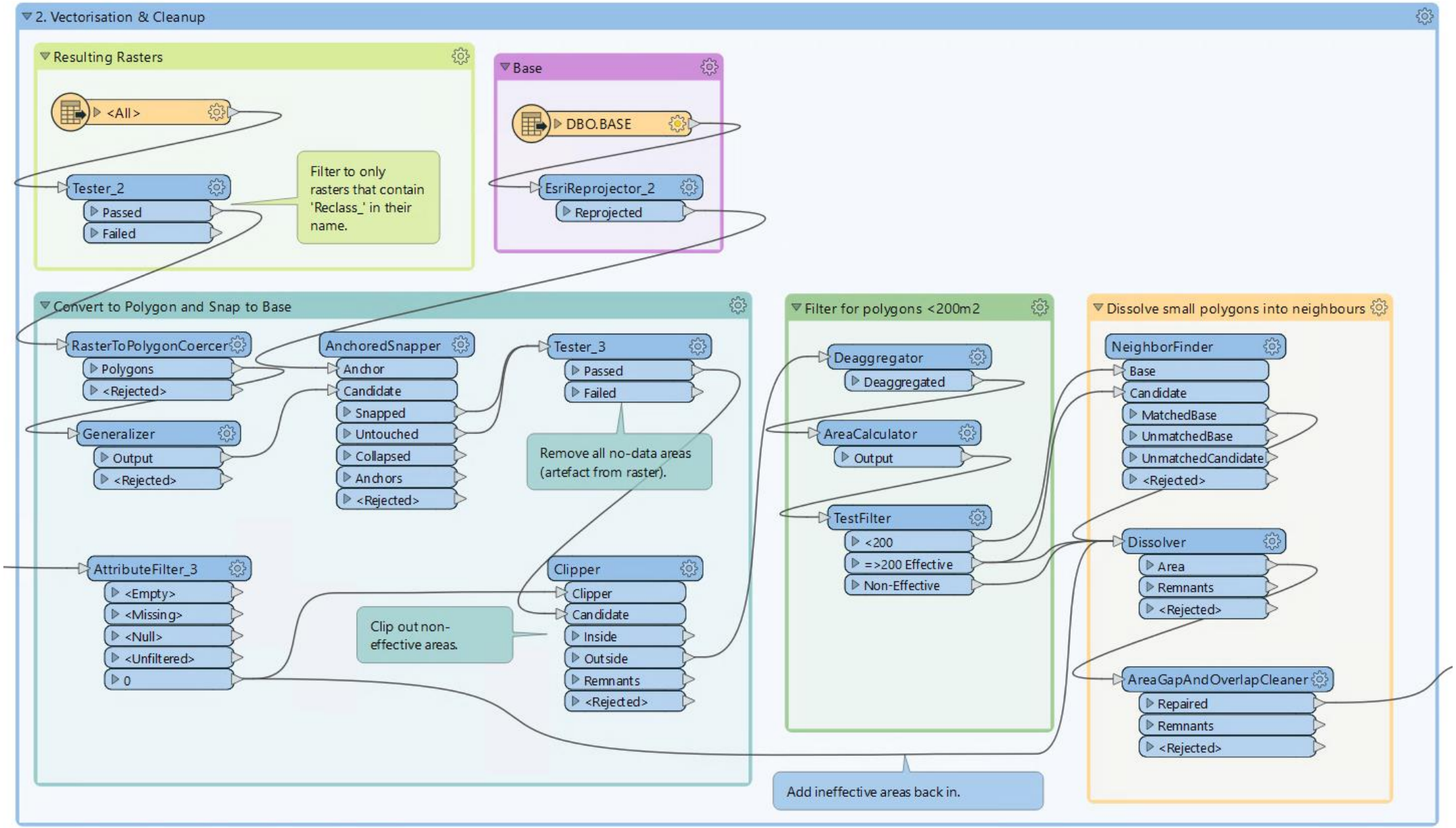









```
def input(self, feature: fmeobjects.FMEFeature):
    """This method is called for each feature which enters the PythonCaller.
    Processed input features can be emitted from this method using self.pyoutput().
    If knowledge of all input features is required for processing, then input features should be
    cached to a list instance variable and processed using group processing or in the close() method.
    """

    #Set volume grid to current feature's LCKEY.
    grid_lyr = arcpy.management.MakeFeatureLayer(self.vol_point_grid, 'grid_lyr')
    arcpy.management.SelectLayerByAttribute(grid_lyr, 'NEW_SELECTION', 'lcid = ' + str(feature.getAttribute('LCKEY')) + ' AND index_ < 9999999')

    print('\t' + str(feature.getAttribute('LCKEY')))


    #Run Natural Neighbour.
    interpolated_raster = arcpy.sa.NaturalNeighbor(grid_lyr, self.value_field, 1)


    #Setting non-log cat areas to No Data.
    print('\t\tClipping to Log Cat Area.')

    eff_lyr = arcpy.management.MakeFeatureLayer(os.path.join(FME_MacroValues['Eff_Area_FGDB'], 'EffectiveArea'))
    arcpy.management.SelectLayerByAttribute(eff_lyr, 'NEW_SELECTION', 'LCKEY = ' + str(feature.getAttribute('LCKEY')))
    clipped_lyr = arcpy.sa.ExtractByMask(interpolated_raster, eff_lyr)

    #Reclassify to SQ bin by age.

    #Construct remap range based on age.
    fc = os.path.join(FME_MacroValues['Reclass_FGDB'], 'Reclassification')
    fields = ['StartValue', 'EndValue', 'NewValue']
    age = str(int(FME_MacroValues['Measure_Year']) - int(feature.getAttribute('PYEAR')))
    where = 'Age = ' + age

    print('\t\tReclassifying using SQ bins for age ' + age + '.')

    #Construct remap table from feature class.
    with arcpy.da.SearchCursor(fc, fields, where) as cursor:
        remap_list = [[row[0], row[1], row[2]] for row in cursor]
        remap_table = arcpy.sa.RemapRange(remap_list)
```


2. Vectorisation & Cleanup
Resulting Rasters
<All>
Tester_2
Passed
Failed
Filter to only rasters that contain 'Reclass_' in their name.
Base
DBO.BASE
EsriReprojector_2
Reprojected
Convert to Polygon and Snap to Base
RasterToPolygonCoercer
Polygons
<Rejected>
Generalizer
Output
<Rejected>
AnchoredSnapper
Anchor
Candidate
Snapped
Untouched
Collapsed
Anchors
<Rejected>
Tester_3
Passed
Failed
Remove all no-data areas (artefact from raster).
AttributeFilter_3
<Empty>
<Missing>
<Null>
<Unfiltered>
0
Clipper
Clipper
Candidate
Inside
Outside
Remnants
<Rejected>
Clip out non-effective areas.
Filter for polygons <200m2
Deaggregator
Deaggregated
AreaCalculator
Output
TestFilter
<200
=>200 Effective
Non-Effective
Dissolve small polygons into neighbours
NeighborFinder
Base
Candidate
MatchedBase
UnmatchedBase
UnmatchedCandidate
<Rejected>
Dissolver
Area
Remnants
<Rejected>
AreaGapAndOverlapCleaner
Repaired
Remnants
<Rejected>
Add ineffective areas back in.

Summary

Core GIS Workflows

Worth the investment!

New Horizons

A new way of thinking about automation and data workflows.

Enhancing Existing Tools with FME

It doesn’t have to fully replace your whole workflow, but it sure can help.

FME in the Forest

John Cannon

the creative fibre group

One Forty One