Custom Civil 3D Subassemblies – Why Would I Need That??

Matthew Dalton

Digital Engineering Lead, WSP

AUTODESK UNIVERSITY

About the speaker

Matthew Dalton


wsp

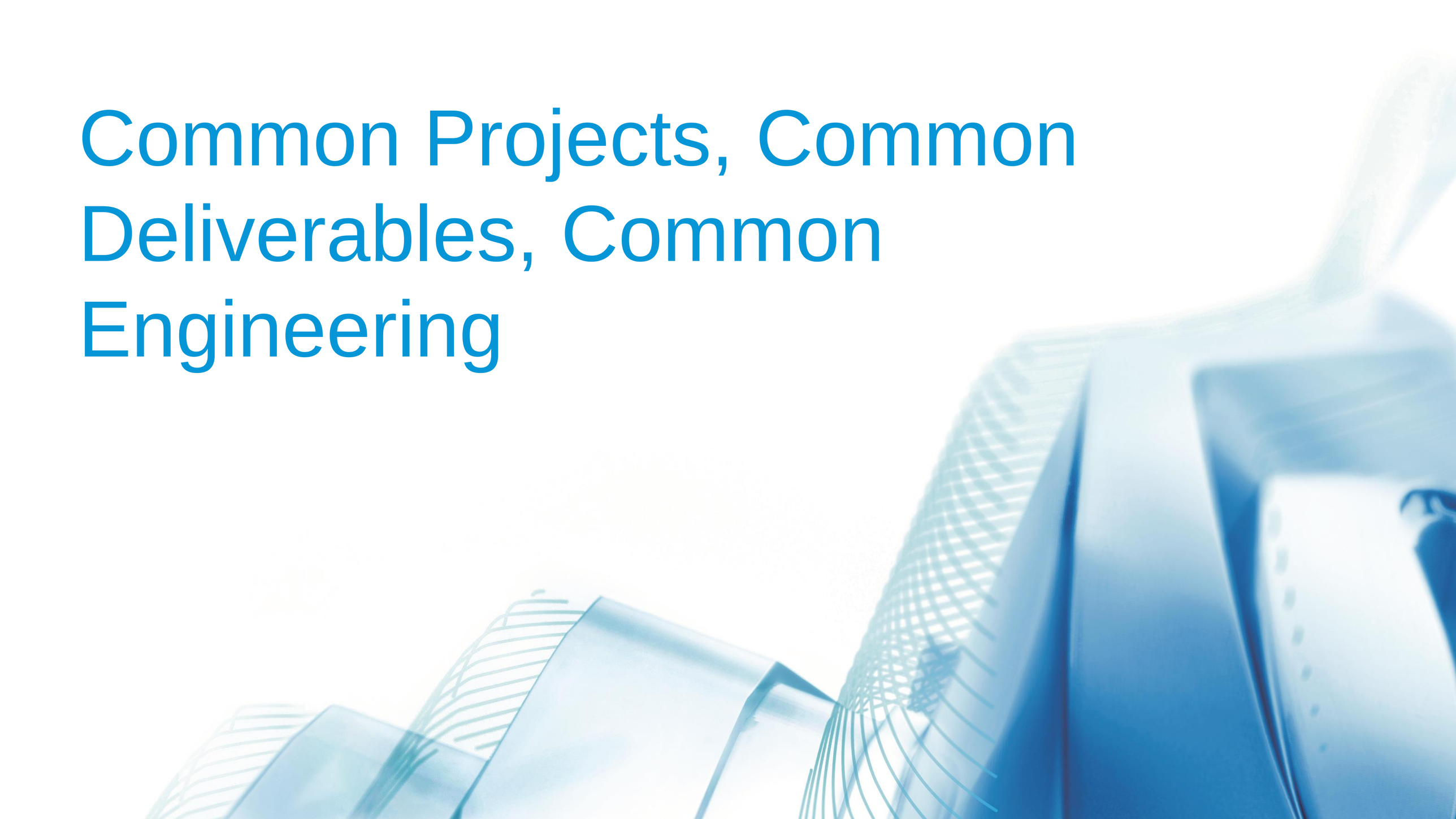
WSP in the UK

+8,000 staff

Breakdown

- Common projects, Common Deliverables, Common Engineering
- Civil 3D ‘out of the box’ and Common project issues
- Custom Subassemblies
  - The Basics – Point, link and shape codes
  - Next Step – Keeping it simple!
  - Getting Clever – Linking it all up!
  - The Future

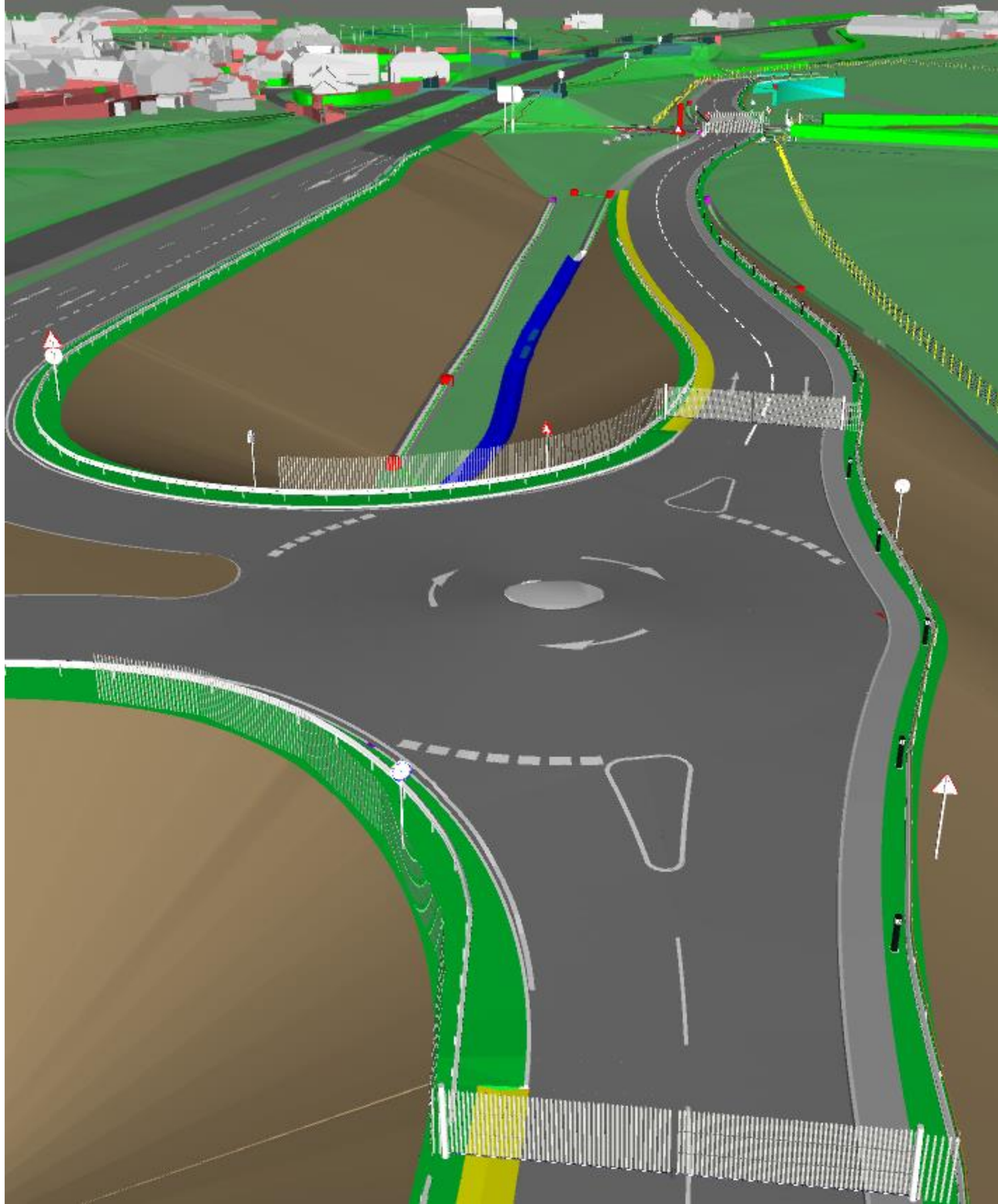
Common Projects, Common Deliverables, Common Engineering



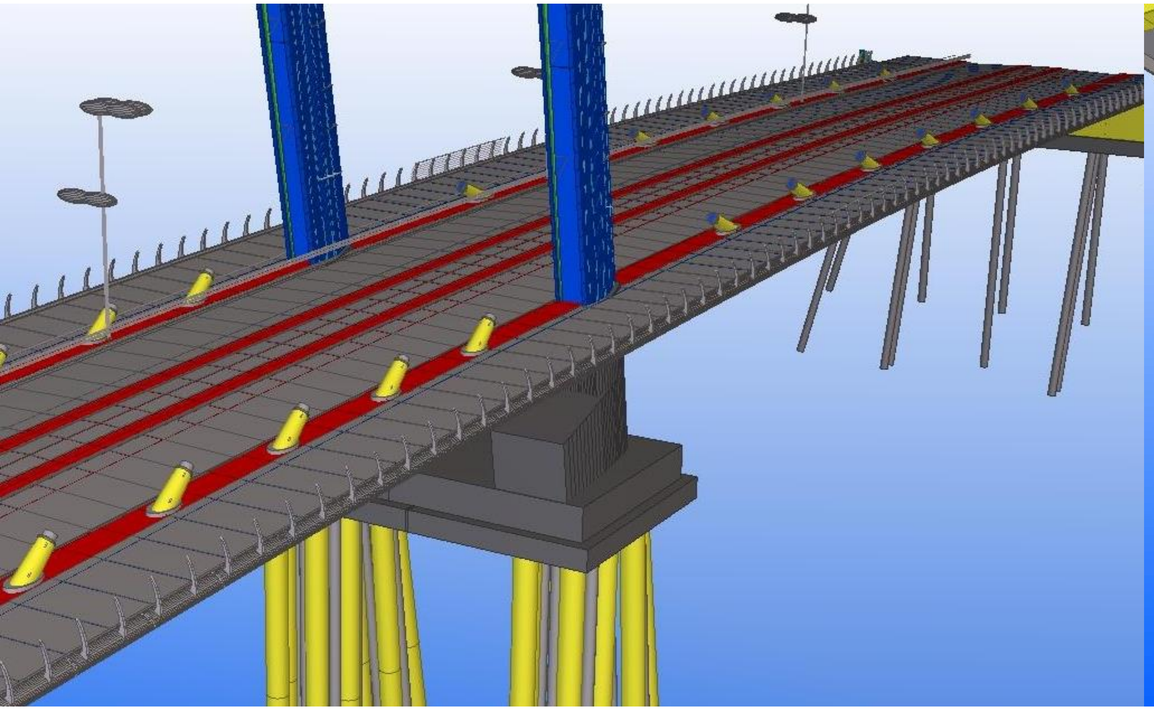
Common Projects, Common Deliverables, Common Engineering

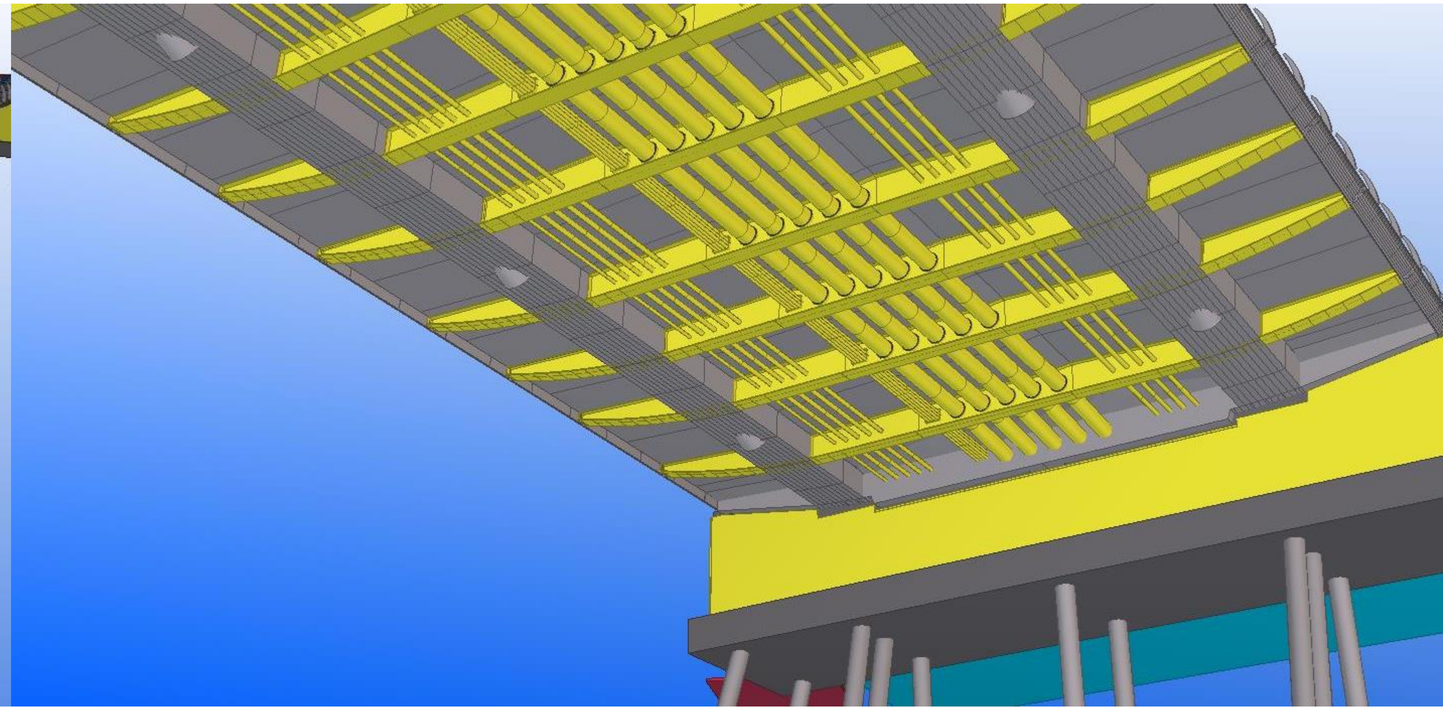
- Traditional CAD drawings

- Plan view ‘string’ output files (corridor feature lines, grading feature lines, alignments, drainage networks etc.)



Common Projects, Common Deliverables, Common Engineering





- 3D Solids for use in clash detection and in a federated Project Information Model (PIM)
- Setting out information (feature lines, points)

Common Projects, Common Deliverables, Common Engineering

- GIS outputs (to link to client Asset Information Models – Digital Twins)

- Asset Data

Civil 3D ‘out of the box’ and Common Project Issues

High Speed Rail 2 (HS2)

High Speed Rail 2 (HS2)

NETWORK

The total HS2 network will be around 330 miles connecting the North and South of the country.

TOP SPEED

HS2 trains will run up to 400kph, faster than any other European high speed trains

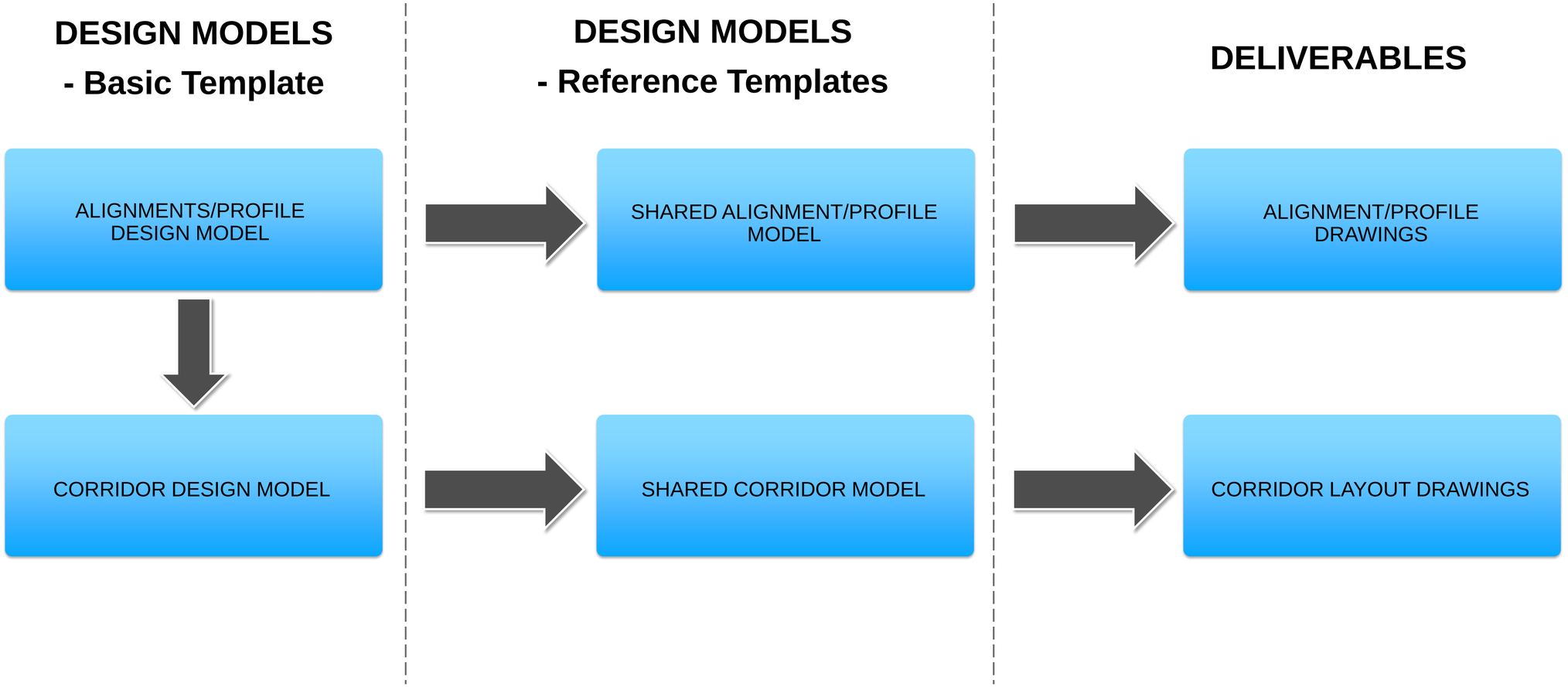
ECONOMY

The network is expected to generate benefits of £59 billion (including wider economic impacts)

EMPLOYMENT

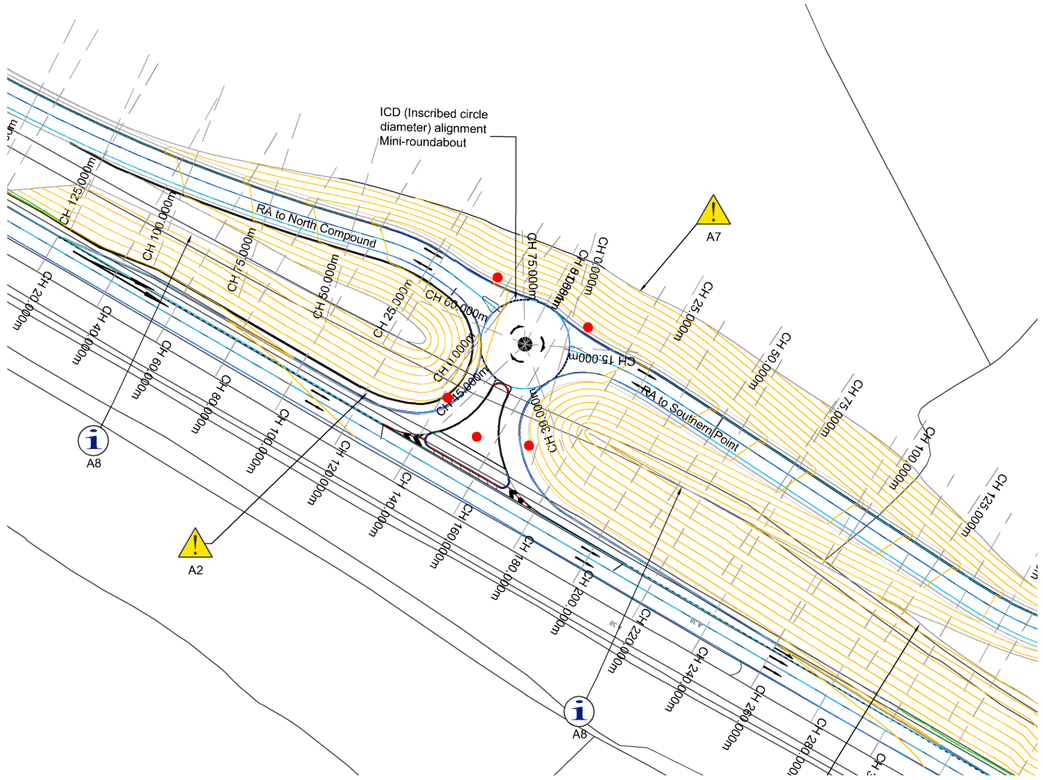
The whole country will benefit from HS2, with Phase 1 predicted to support around 40,000 jobs.

Model breakdown - Example



Civil 3D ‘out of the box’ and Common Project Issues

Checking, re-working, second run-throughs of check/review and post-processing all add up to a fairly significant cost, even for those of us with rigorous CAD standards and verification processes!



Custom Subassemblies –

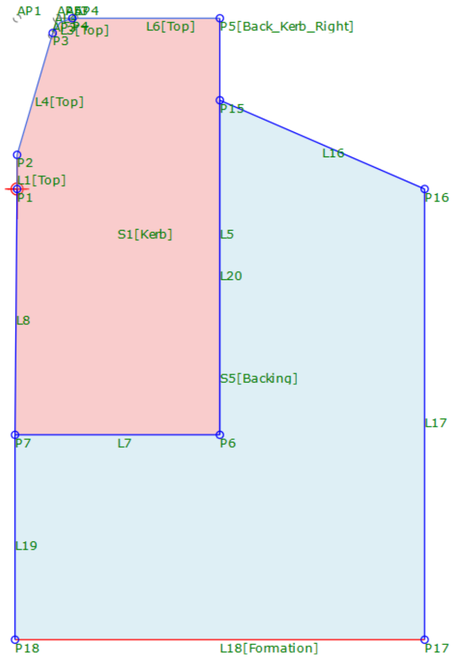
So what can they do that’s beyond conventional?

THE BASICS – Point, Link and Shape Codes

Civil3D subassemblies use three key ‘codes’ in order to produce corridor models and can be varied in their behaviour using ‘Code Set Styles’ within a Civil 3D ‘.dwg’ file.

- Point codes
- Link Codes
- Shape Codes

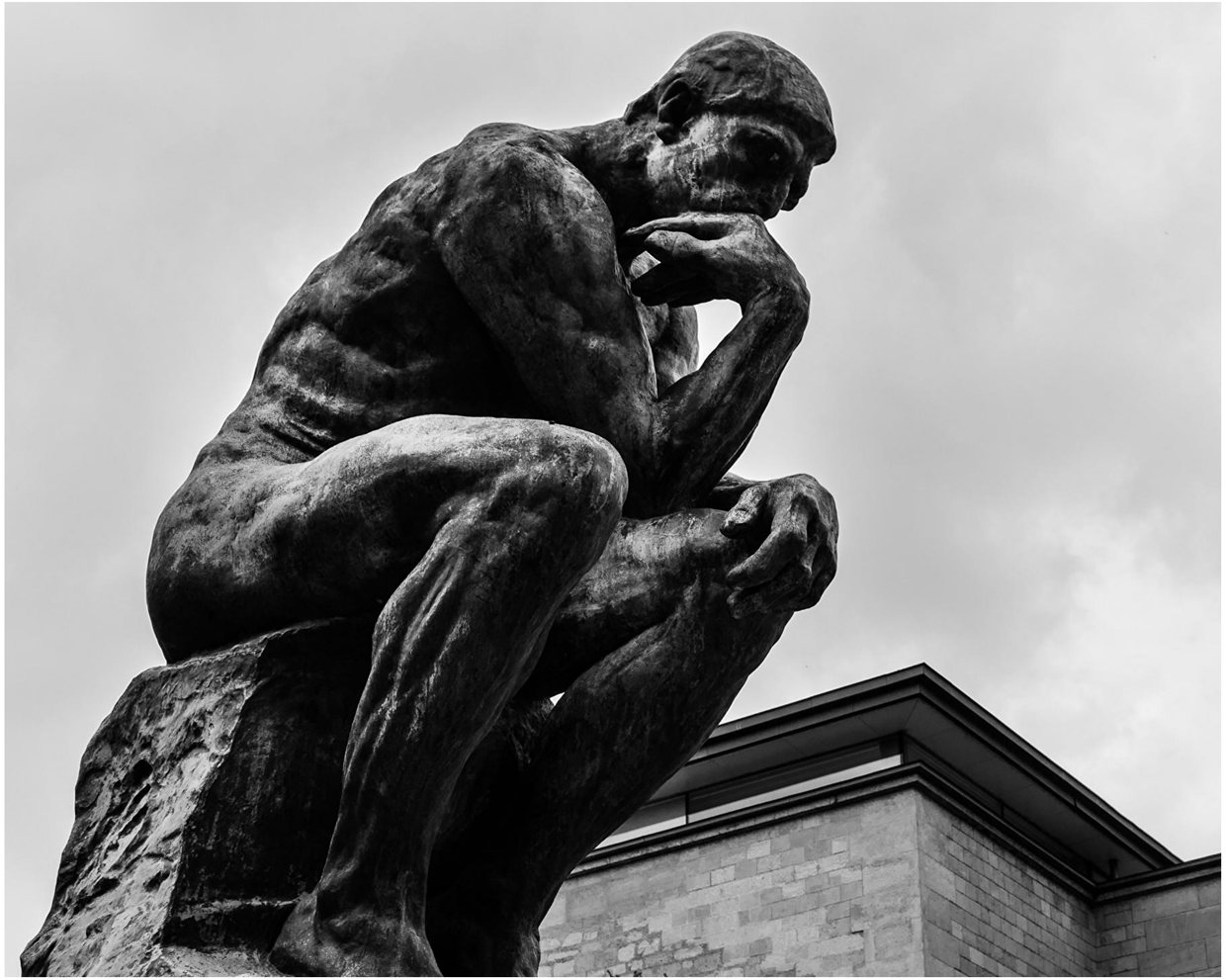
In a traditional ‘out of the box’ subassembly such as those included with a country kit, these are all manual entry.



Point Codes

How do we reduce Point code related errors?

- Typos
- Inside and Outside codes used
- Overlapping featurelines
- Etc!!



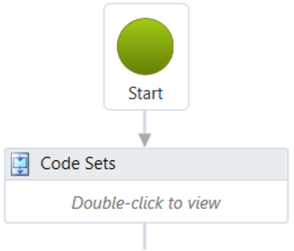
Point Codes

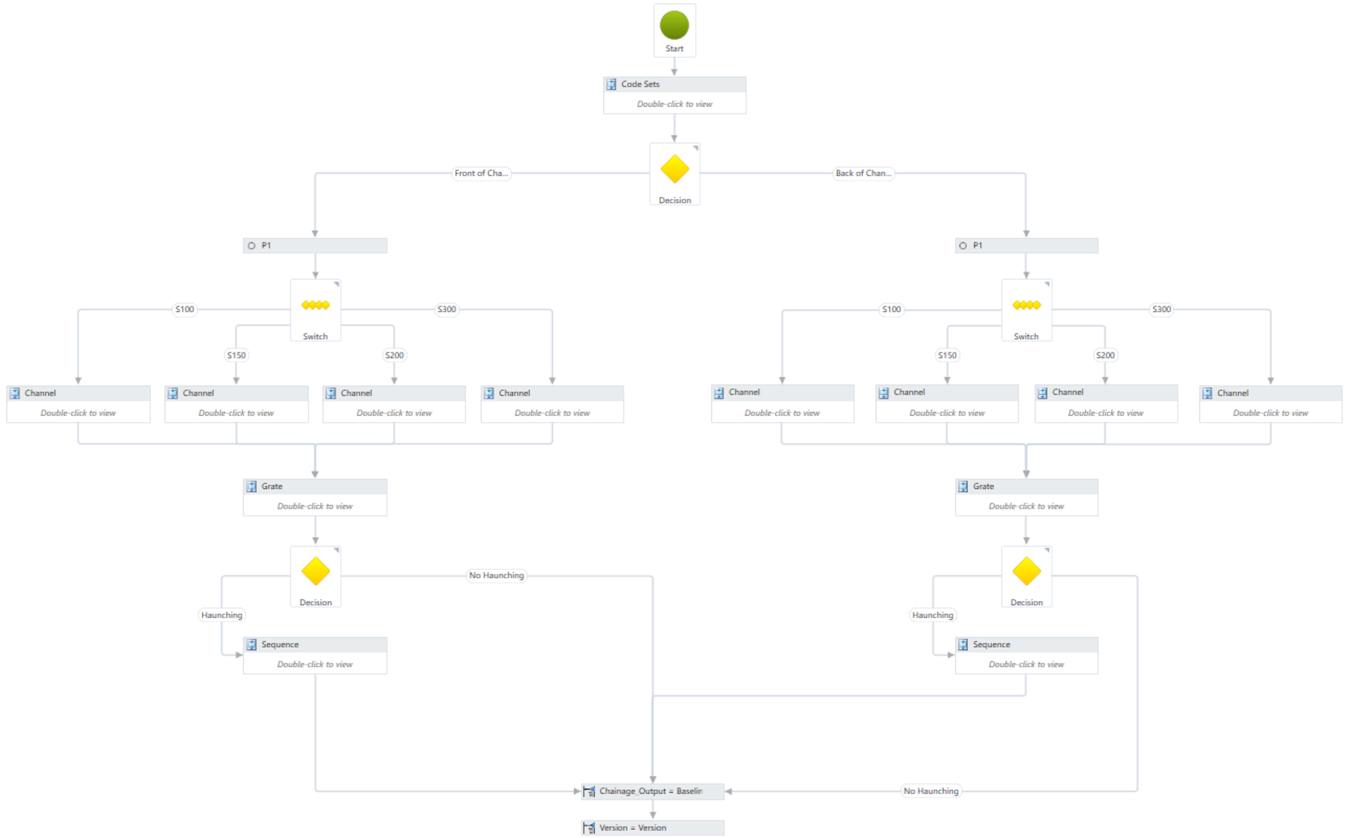
THE SOLUTION

- Hard-code where possible
- Include ‘side’ information
- Pre-define point codes that can be used in the subassembly
- Choose which of those codes was applicable
- No manual entry = No typos!
- Be common across all subassemblies

The Layout:

- Keep it tidy
- Make it readable
- Leave it so others can understand
- Organised, locked elements!





Selectable Point Code Values

- How do we have selectable values for our point codes?...

Enumeration Lists!!

Enum Group	Enum Item	Display Name (Shows in Civil3D)
Inside_Codes	Crown	Crown
Outside_Codes	Hardstrip	Hardstrip
Target	InsideShoulder	Inside Shoulder
MySuper	EdgeOfCarriageway	Edge Of Carriageway
Surface_Course	*CreateEnumItem*	
Binder_Course		
Base_Course		
SubBase_Course		
Capping_Course		

Create a parameter for your enumeration list.

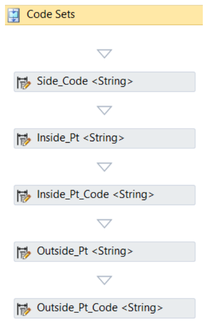
Input/Output Parameters

Name	Type	Direction	Default Value	DisplayName	Description
Side	Side	Input	Left		
Insides	Inside_Codes	Input	Crown	Inside Point Code	Sets inside point code
Outsides	Outside_Codes	Input	EdgeOfCarriagewa	Outside Point Code	Sets outside point code

Then create a string variable and set that to read the parameter name and use the '.value' function.

Variable	
Name	Inside_Pt
Variable Type	String
Default	Insides.value
Miscellaneous	
Comment	

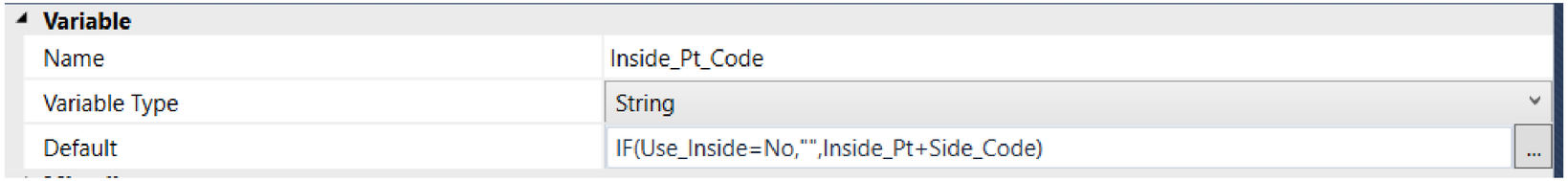
We can then use a second variable to link the side and name together and this is our code variable.



Selectable Point Code Values

And the code we use?

Bring it all together to make it work with a simple ‘Yes/No Parameter…



The Result?

- Easy to use
- Locked Values determined by enumeration lists
- Side information included automatically
- If it’s wrong, the code set misses it and no line appears – Easily checkable!

GETTING CLEVER – Linking it all up!

- What else can we do if that works??
- Link Codes?
- Shape Codes??

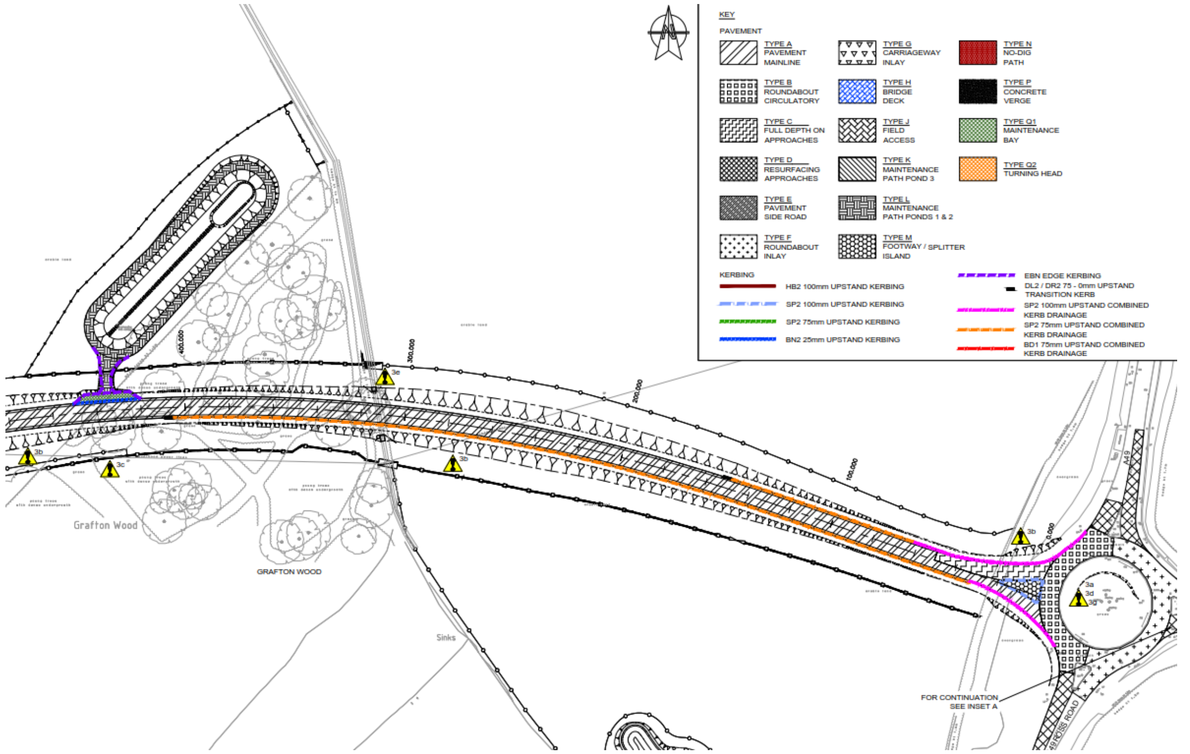
Link Codes

Links

- Simple Option: Set Variable – “Top”

OR…

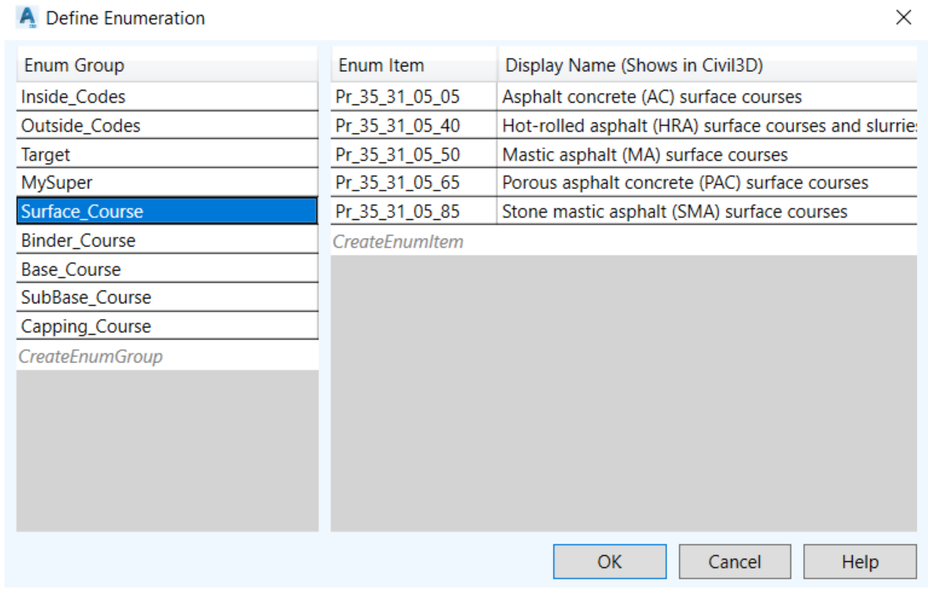
- Use the exact same process of an enumeration list as for point codes..



Pavement Types…

Lets say we want to automate creation of paving layout drawings…

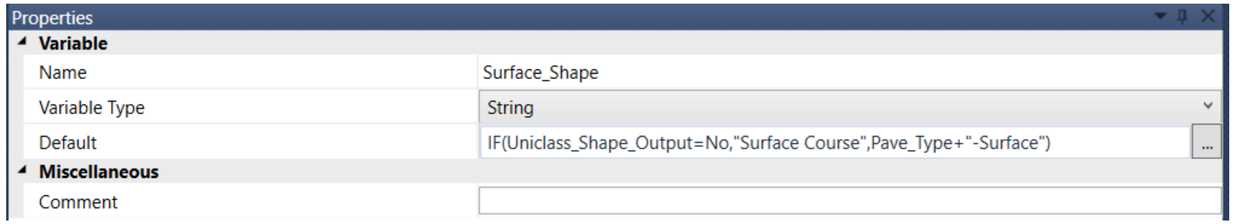
- Enumeration Lists (again!)
- Create a List which reflects your local layering system (the UK uses Uniclass2015)
- Map to link code using string variables (‘.value’ function)



Going Further – Shape Codes

This also all applies to Shape Codes, but we can do more!

- Shape codes may have multiple uses, not just for solid creation
- Use needs to be flexible
- Create Yes/No Parameter – Uniclass_Shape_Output
- Set Shape Code string variable. Needs to reference yes/no parameter and add “-Surface” text (I’ll explain shortly…)



Shape Codes & Parameters – Why the effort?

Now if you’re sat reading this and scratching your head, why would you want to drive layer information to a shape code?

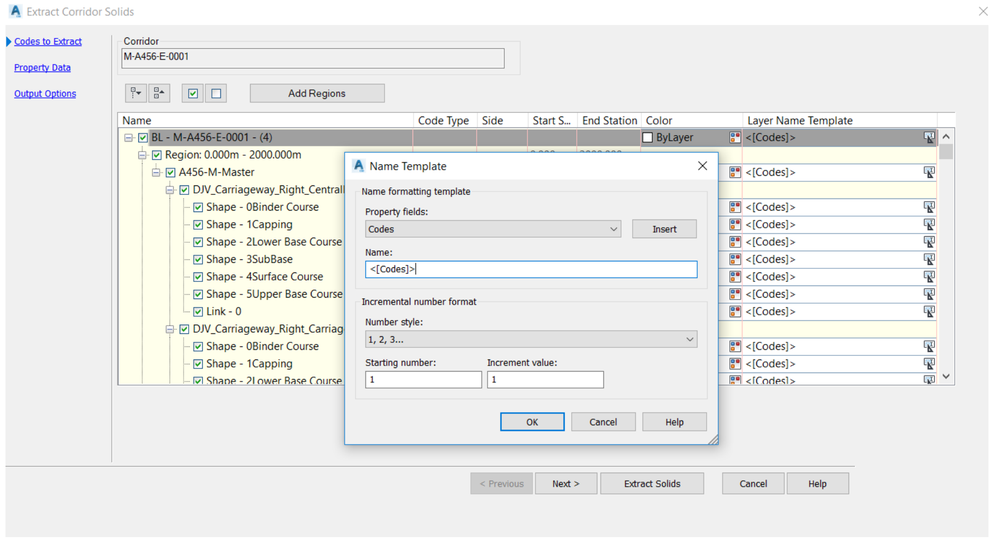
The answer lies in Civil 3D and how it creates 3D solids.

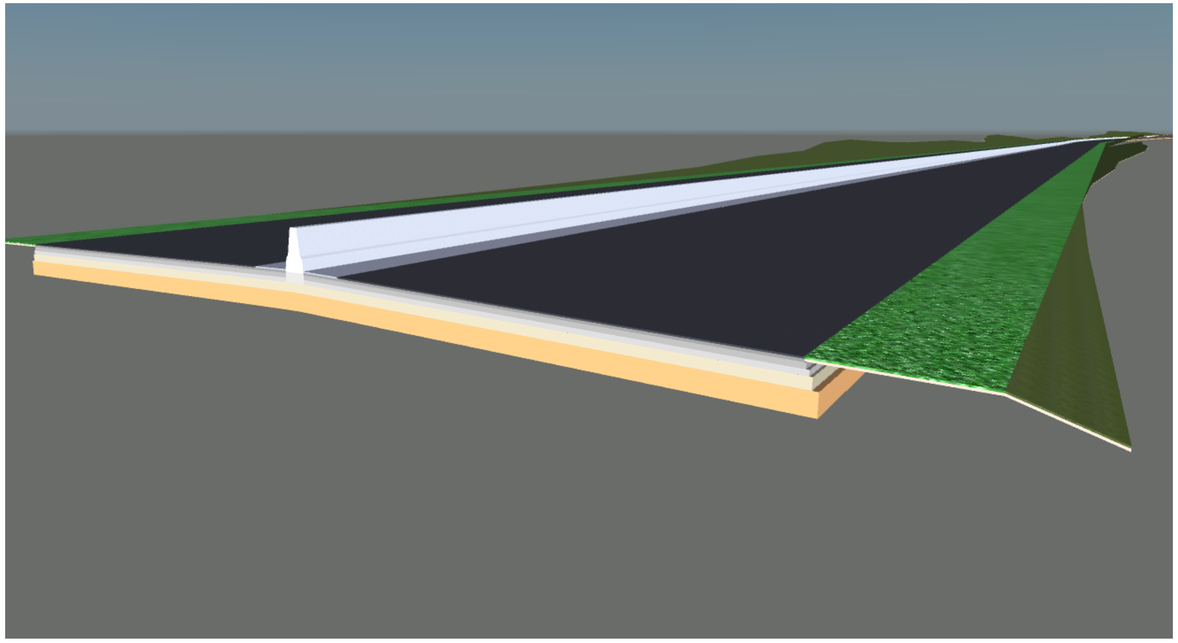
Civil 3D Solid Output

Use your enumerations to drive your outputs!

- Set your name template to ‘Codes’ (or ShapeCodes in older C3D versions)
- Create a reference template for a single featureline only (EdgeofCarriageway)
- Dref your corridor model
- Create solids in that ‘solids only’ model
- Dynamically link the solids to the corridor
- Append to Navisworks

You now have a dynamically updating 3D solid model file, which will maintain its link to your single source of truth!





The Future – Leveraging Your Subassemblies For More!

Time to really stretch things out….

How is a corridor structured in the .net API?

CorridorName

BaselineName

RegionName

Assembly

Subassembly

Subassembly Properties (parameters)

And in the 3D Solids File??

CorridorName

BaselineName

RegionName

Assembly

Unique, Non-repeatable, extractable!!!!

```
- Corridor: TestCorridor - (1) (2319600249968)

  - Solid3d:  (2319600253072)
    *A1:
      CorridorName: TestCorridor - (1)
      CorridorDesc:
      BaselineName: BL - M-Mainline Dual Carriageway-1 - (2)
      HorizontalBaseline: M-Mainline Dual Carriageway-1
      VerticalBaseline: M-Mainline Dual Carriageway-1 - Training_Topo - SP 2
      RegionName: RG - TestAssembly - (1) - (2)
    *A3:
      CodeName: Topsoil_Shape, Grass
      Side: Left
      RateItem:
      ClassificationCode:
      AssemblyName: TestAssembly - (1)
      AssemblyStartStation: 0.000m
      AssemblyEndStation: 6711.992m

    - Assembly: TestAssembly - (1) (2319600249824)

      - Subassembly: Subassembly3 (2319600249952)
        Subassembly Properties:
          Verge Grade: -0.025
          Sub-Surface Depth: 0.3
          Topsoil Depth: 0.15
          Verge Width: 2.5
          Formation Link Code: Formation
          Inside Point Code: Verge_In
          Outside Point Code: Verge_Out
          Subsoil Shape Code: Subsoil_Shape
          Subsoil Top Link Code: Subsoil_Top_Link
          Topsoil Base Link Codes: Topsoil_Base_Code
          Top Link Codes: Top, Verge
          Topsoil Shape Code: Topsoil_Shape
          Verge Sub Material: Sub-Soil
          Side: Left
          Verge Top Material: Grass
```

Have at me!!

AUTODESK®

Make anything™