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

Matthew Dalton
Digital Engineering Lead, WSP
About the speaker

Matthew Dalton
in the UK

+8,000 staff

SCOTLAND
475 people

NORTHERN IRELAND
80 people

NORTH OF ENGLAND
2,100 people

MIDLANDS
1,100 people

WALES & SOUTH WEST
1,250 people

LONDON & SOUTH EAST
3,300 people
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)
The total HS2 network will be around 330 miles connecting the North and South of the country.

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

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

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

**DESIGN MODELS**
- Basic Template
  - ALIGNMENTS/PROFILE DESIGN MODEL
  - CORRIDOR DESIGN MODEL

- Reference Templates
  - SHARED ALIGNMENT/PROFILE MODEL
  - SHARED CORRIDOR MODEL

**DELIVERABLES**
- ALIGNMENT/PROFILE DRAWINGS
- CORRIDOR LAYOUT DRAWINGS
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!
• How do we have selectable values for our point codes?

**Enumeration Lists!!**

<table>
<thead>
<tr>
<th>Enum Group</th>
<th>Enum Item</th>
<th>Display Name (Shows in Civil3D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inside_Codes</td>
<td>Crown</td>
<td>Crown</td>
</tr>
<tr>
<td></td>
<td>Hardstrip</td>
<td>Hardstrip</td>
</tr>
<tr>
<td></td>
<td>InsideShoulder</td>
<td>Inside Shoulder</td>
</tr>
<tr>
<td></td>
<td>EdgeOfCarriageway</td>
<td>Edge Of Carriageway</td>
</tr>
</tbody>
</table>

Create a parameter for your enumeration list.

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

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

Selectable Point Code Values
GETTING CLEVER – Linking it all up!

• What else can we do if that works??

• Link Codes?

• Shape Codes??

what else?
Link Codes

Links

• Simple Option: Set Variable – “Top”

OR...

• Use the exact same process of an enumeration list as for point codes.
Pavement Types…

*Let's 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)
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!
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!!!!
Have at me!!