



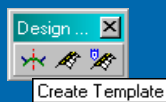
# Typical Sections - Objectives

- Design Roadway Workflow Introduction
- Template Modeling Fundamentals
- Creating a Typical Section Library
- The Create Template Interface
- Create Template dialog - Nuts and Bolts
- Section Library Fundamentals
- Template Creation General Workflow
- Common Section Component Types

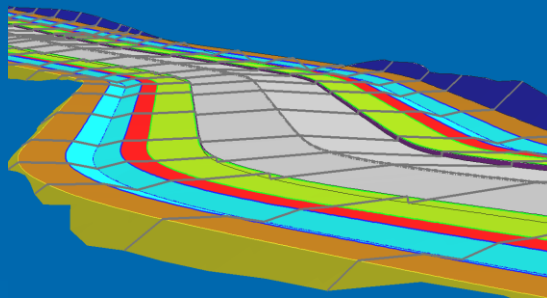
© 2010 Zen Engineering / Mark S. Ditko

Page 7-1  
04 Aug 10

## Corridor Design



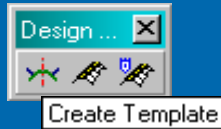
- Once the geometry is complete modeling is only a few steps away.
- **Typical Sections** are combined with the existing **Surface** and new **Geometry** to create a roadway DTM model



© 2010 Zen Engineering / Mark S. Ditko

Page 7-2  
04 Aug 10

# Design Roadway Toolbar



- The **Create Template** command is found on the *Design Roadway* toolbar and under **Modeler** on the Main Menu bar.
- This tool is also used to create *Cut & Fill* sloping conditions

## Template Basics



- Templates are constructed by placing points to form a section.
- These points define locations like the Centerline, Gutter, Top of Curb, Sidewalk edge, Ditch flow lines and Cut and Fill catch points

# Breakline Creation

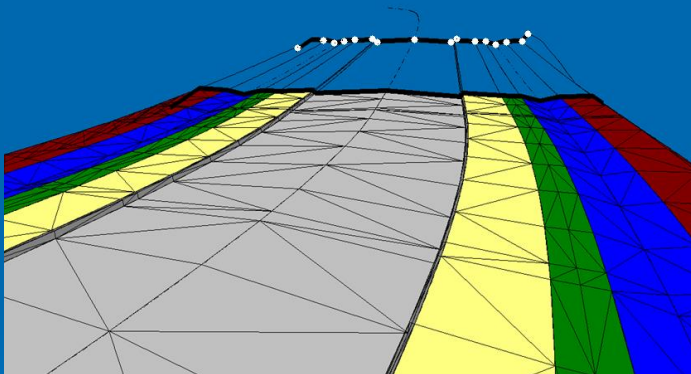


- Each Point on the Template constructs a breakline as it travels along the path defined by the Horizontal and Vertical alignments
- These breaklines are stored in a new surface

© 2010 Zen Engineering / Mark S. Ditko

Page 7-5  
04 Aug 10

# Surface Construction



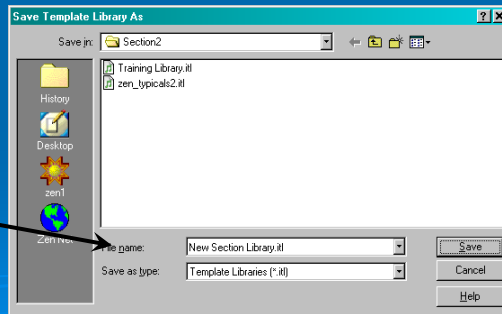
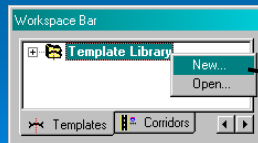
- After the breaklines are generated the surface is triangulated which forms the DTM

© 2010 Zen Engineering / Mark S. Ditko

Page 7-6  
04 Aug 10

# Section Library

- Right-click in the Workspace Bar on **Template Library** > **New...** to create a new Template Library
- The *Template Library* holds individual Templates or Sections
- Only 1 Library can be open at a time

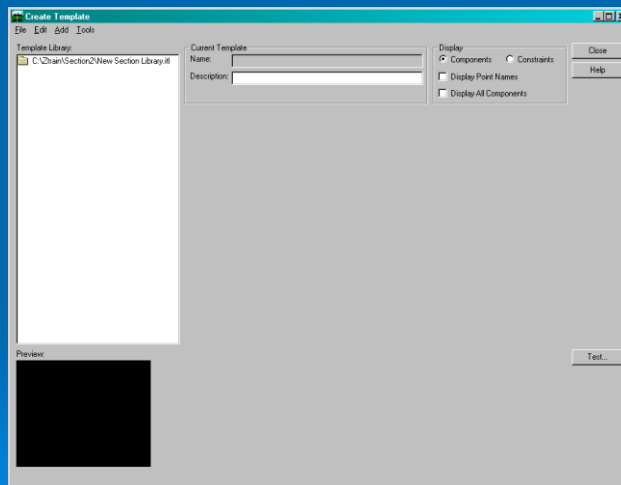


© 2010 Zen Engineering / Mark S. Ditko

Page 7-8  
04 Aug 10

# Template Library - New

- Brand new **Template Libraries** are completely empty

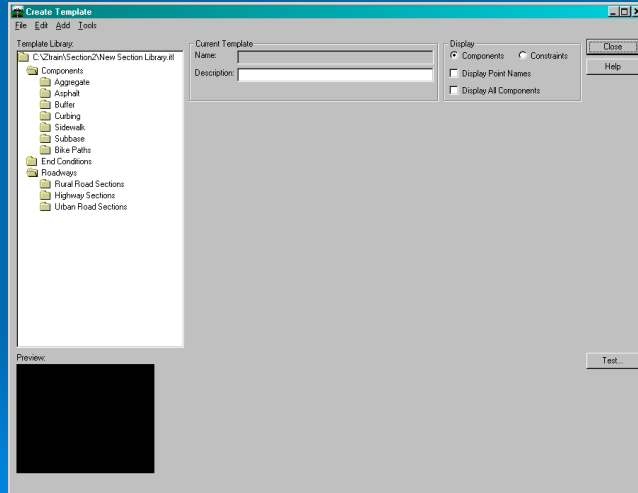


© 2010 Zen Engineering / Mark S. Ditko

Page 7-9  
04 Aug 10

# Template Library - Folders

- A folder structure is created first
- It can be as simple or as detailed as necessary
- The folders are for your organization

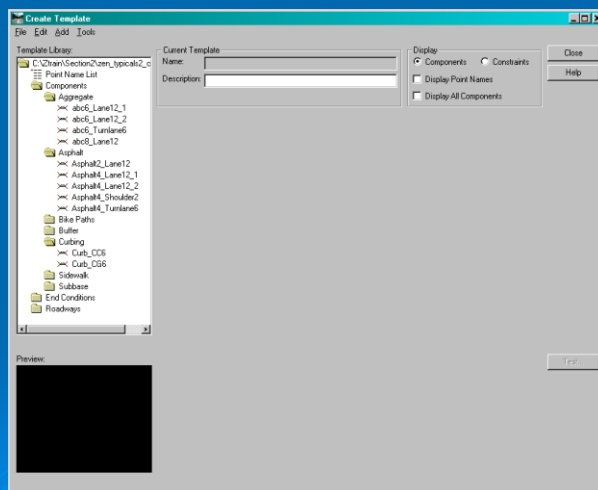


© 2010 Zen Engineering / Mark S. Ditko

Page 7-10  
04 Aug 10

# Template Library - Sections

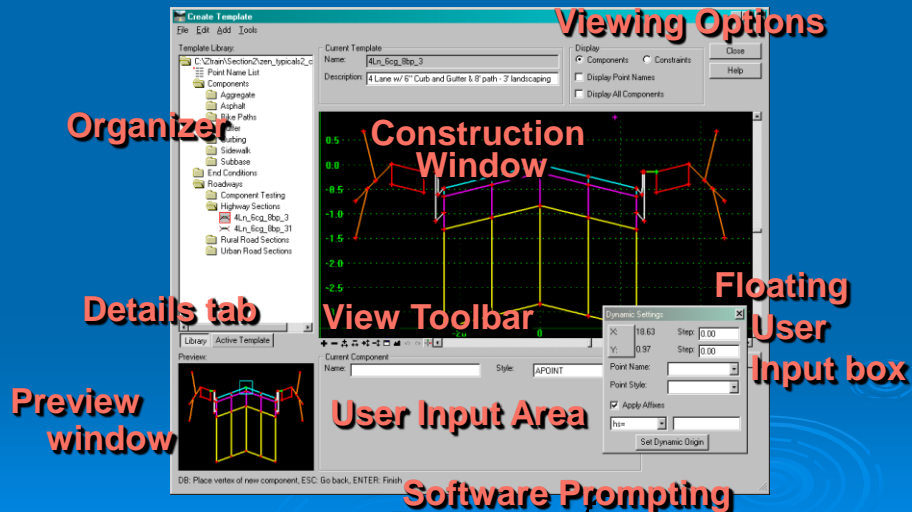
- Section Names can be created within the new folder system
  - One section name can be created then 'built'
  - Many names can be made
  - A good naming convention is important



© 2010 Zen Engineering / Mark S. Ditko

Page 7-11  
04 Aug 10

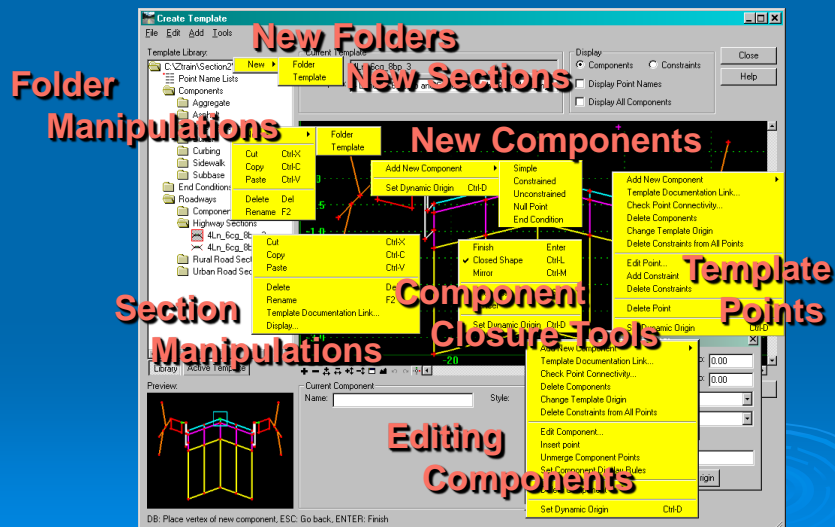
# The Create Template Interface



© 2010 Zen Engineering / Mark S. Ditko

Page 7-12  
04 Aug 10

# A Right-click World

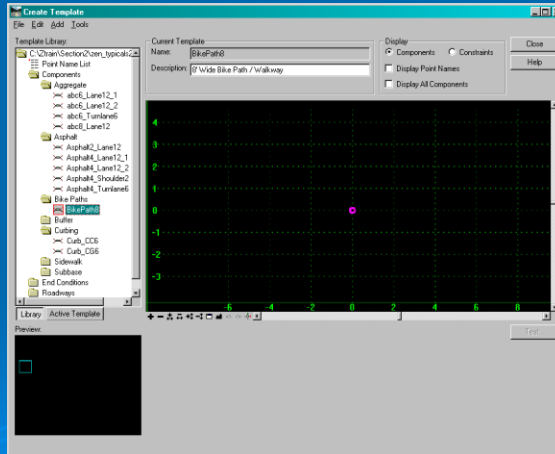


© 2010 Zen Engineering / Mark S. Ditko

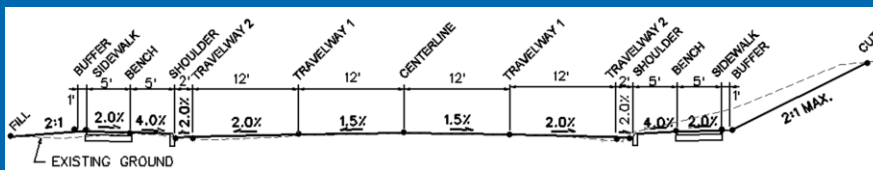
Page 7-13  
04 Aug 10

## Section Construction

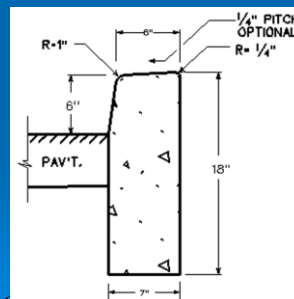
- Once any folders & section *system* is developed the Section is then created
- Sections are created using 2 basic techniques

Page 7-14  
04 Aug 10

# Sections – Fundamentals 1



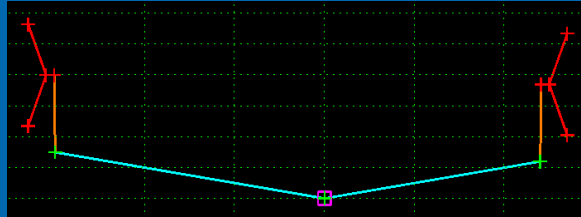
- Points are placed to precisely define the section
- Make sure you know the slope, width and other dimensional criteria before beginning
- Sketch it out



© 2010 Zen Engineering / Mark S. Ditko

Page 7-15  
04 Aug 10

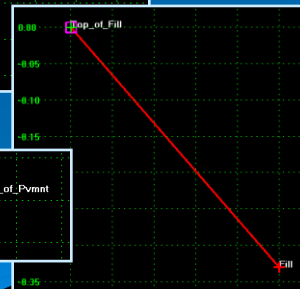
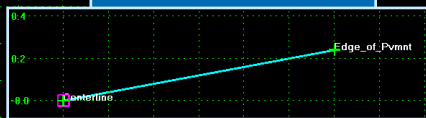
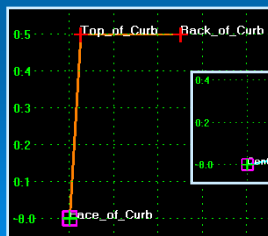
## Sections – Fundamentals 2



- Sections can be built using two basic techniques:
  - From Scratch (Newly constructed within the section - point by point ...like InRoads V8.05 and earlier.)
  - Drag & Drop (Build a section from precreated 'parts')

## Sections – Fundamentals 3

- Building Sections from 'parts' can be a faster construction technique
- Section 'parts' are called **Components**

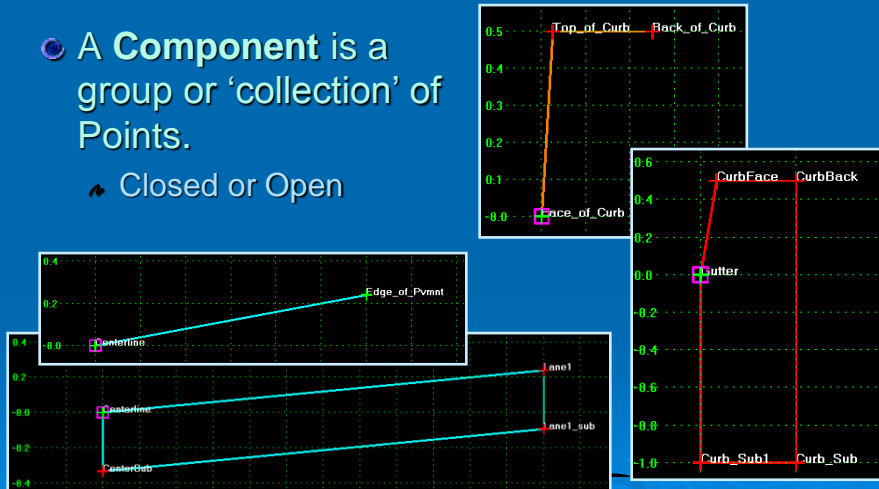




## Sections – Fundamentals 4

- A **Component** is a group or 'collection' of Points.

➤ Closed or Open

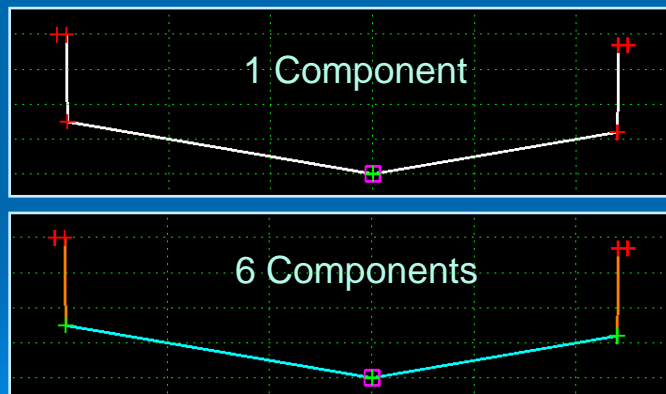


© 2010 Zen Engineering / Mark S. Ditko

Page 7-18  
04 Aug 10

## Sections – Fundamentals 5

- A **Component** can encompass the entire body of a Section, or can be smaller individual pieces.

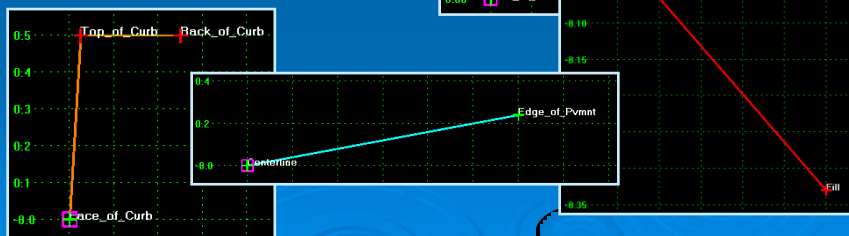


© 2010 Zen Engineering / Mark S. Ditko

Page 7-19  
04 Aug 10

## Sections – Fundamentals 6

- A good standard section library will make it easier to build the exact template that you need for your project



© 2010 Zen Engineering / Mark S. Ditko

Page 7-20  
04 Aug 10

## Show Section Construction

- DEMONSTRATE 'Drag & Drop'

...

© 2010 Zen Engineering / Mark S. Ditko

Page 7-21  
04 Aug 10

# Terminology Clarification

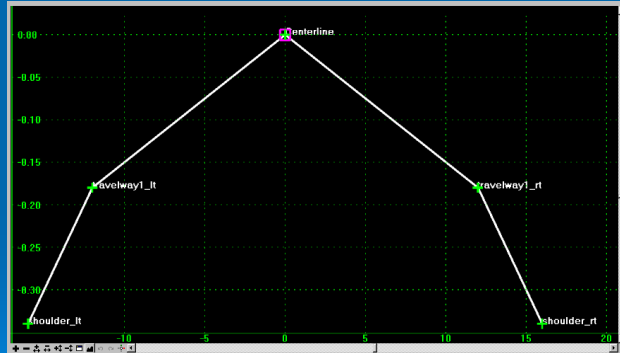
- To assist in keeping the subject of Typical Sections clear, the following terms should be understood
  - ♣ Points
  - ♣ Components
  - ♣ Templates

## Terminology Concepts 2

- **Points** are used to build a Component
- For simplicity, a **Component** can be considered a smaller piece of a section ...or 'area of interest'
- A **Template** usually consists of several Components put together to form a full section
- Within InRoads, a Component is sometimes referred to, and used interchangeably with a Template (for instance the **Apply Template** command, and in the **Create Template** tool a single Component is given a 'Template Name'.)

# Section Points Expanded

- All DTM data contain three characteristics
  - ♣ Unique **Name**
  - ♣ Display **Style**
  - ♣ **Type**
- Section points are **named**
- Points are assigned a display **Style**
- All section points become **breaklines**



© 2010 Zen Engineering / Mark S. Ditko

Page 7-24  
04 Aug 10

# Section Characteristics

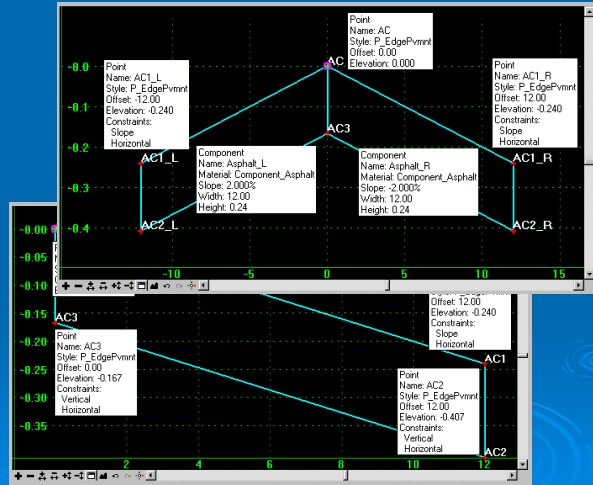
- User required info during section creation
  - ♣ **Templates:**
    - Template **Name** – Used to identify the Section
  - ♣ **Points:**
    - Point **Name** – The Feature Name in the surface
    - Point **Style** – Defines DTM Feature intelligence
  - ♣ **Components:**
    - Component **Name** – ‘Categorize’ these ‘pieces’
    - Component **Style** (same as Component “Material”) – Used for display of the Component as well as defining its material for quantity take-offs

© 2010 Zen Engineering / Mark S. Ditko

Page 7-25  
04 Aug 10

# Informational Feedback

- Float the mouse pointer over various pieces of 'graphics' to provide feedback

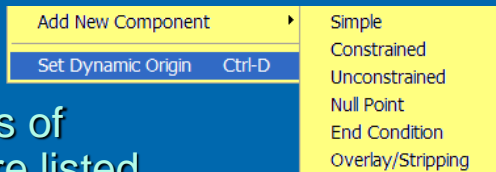


© 2010 Zen Engineering / Mark S. Ditko

Page 7-26  
04 Aug 10

## Component – Types

- At creation, 6 different types of Components are listed  
(In reality there are 3 key types)



- Rigid** or **Constrained**
      - Simple** (Simple)
      - Complex** (Constrained)
    - Free** or **Unconstrained** (Unconstrained)
    - End Conditions**

© 2010 Zen Engineering / Mark S. Ditko

Page 7-27  
04 Aug 10

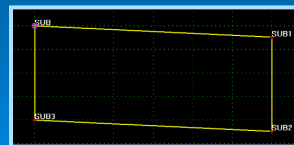
# Component – Constrained 1

- **Rigid** or **Constrained**: All points of the component are controlled and driven by “the One”. The movement of “the One” forces the movement of the others.

There is a ‘Parent - Child’ relationship between “the One” and the “others”. The “others” cannot influence “the One”.

- **Simple**: Auto-parallel defined by thickness, width & slope.
  - Pavement section, Sidewalk slab, Asphalt layer, Aggregate Base, Compacted Subgrade layer... and so on.
  - Simple components are always closed parallelogram shapes

“The One”



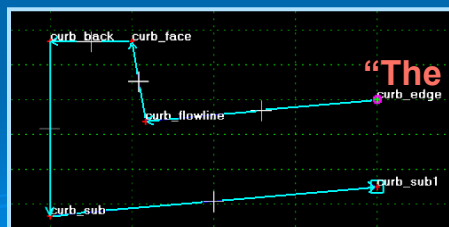
© 2010 Zen Engineering / Mark S. Ditko

Page 7-28  
04 Aug 10

# Component – Constrained 2

- **Rigid** or **Constrained**: All component points are controlled and driven by “the One”. “The One” forces the movement of the others.

- **Complex**: Flexible user-defined closed shapes or open line work.
  - Curb & Gutter, Medians, Barriers, shoulders...



© 2010 Zen Engineering / Mark S. Ditko

Page 7-29  
04 Aug 10

## Component – Other Types

- **Free or Unconstrained:** Points of the component act independent of one another and do not contain any Parent / Child relationships. Unconstrained can be either open or closed.
- **Null Point:** A specialized 'location' that can be referenced to other points in order to provide unique controls
- **End Conditions:** Define the side slope conditions intended to intercept existing
- **Overlay / Stripping:** Specialized component

## Typical Sections - Summary

- Before Sections can be constructed, a **Template Library** (ITL) must be created or loaded.
- Only **one Library** can be loaded at a time.
- An Organization & Naming system is important for both the Folder structure as well as the Sections
- Creating a **new template** in the Library is done through the Create Template dialog box.
- Understand the role of **Points, Components**, and the **Templates** within the section library.
- Different types of components are used to build the section pieces, most commonly **Rigid, Free** and **End Conditions**

# Typical Sections – Part 2

## ● QUESTIONS ??

...

# Typical Sections – Review 1

- What is a **Component**?
- What are the basic **Component Types**?
- Exactly what IS a **Constraint** anyway?
- Give an example or two of a **Parent – Child** “constraint” relationship that might exist between points in a component
- Explain what the **Red / Yellow / Green Template Points** are indicating ...why care?
- How do you **relocate a point** in a component?
- How does a template have to be constructed if you expect that template to **be able to transition**?



## Typical Sections – Review 2

- What is **Mirroring** and **Reflecting** for?
- What does the **Dynamic Settings** dialog do?
  - Let's take a look at the End Conditions 'DS' dialog
- When would you, or would you not **Apply Affixes**?
- What is the **Check Point Connectivity** tool all about? Why do it?
  - Show  $4+4=??$
- What are **End Condition Priorities**?

## Typical Sections – More

- All the Template Points in a Typical Section get placed into a **single design DTM** by Default
- Point Properties - **Alternate Surface** capability
- **Active Template** tab in the Organizer
- There is a **Template Library Organizer** to copy Sections from one library to another
- **Tools> Apply Feature Name Override...**
- **Advanced** Templating consists of:
  - Complex Constraints (Constraint Types & Connections)
  - “Parent Components” & Component Display Rules