

## TECHNICAL TIP ~ InRoads DTM Conflicts, Part 5

### DISSECTING THE “ERROR LOG” ... AND BEYOND

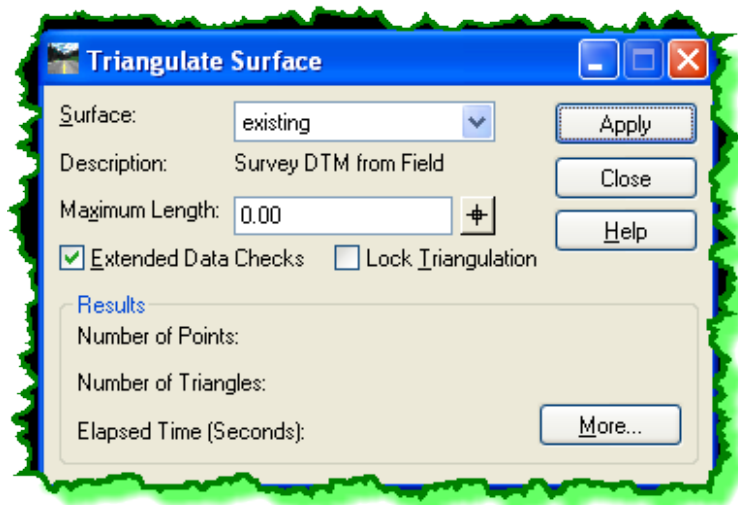
Before I start I want to make sure that you know that I’m writing this technical information based on using InRoads V8i, SS2, (V08.11.07.615). But I would say that even though most of the content in these tips applies to all of the recent V8 versions, you should be alert to the fact that things can change from version to version, and even from a minor version to a minor version.

HHHHhhhhhhh... [Big exhale]

Okay, I have to be honest; this is where I start to feel like Alice in Wonderland after having gone down too deeply into the rabbit hole.

In this tip I’m going to attempt to clarify some terms and paint a bigger picture regarding ‘errors’. So let me begin to structure this a bit.

First there is the **Triangulate Surface** command: (I’m assuming that all of you know how this command functions.)



I mentioned earlier that the **Triangulate Surface** “Error Log” shows two different types of “errors”. (Remember that this *Error Log* report will only appear if the **Extended Data Checks** option is toggled on, and the InRoads **Report Lock** is also in the on position.)

There are two basic types of ‘errors’ that you’ll run into when working with surfaces.

- 1) **Mismatched Elevations** – I consider these as the “real errors” and the ones that I focus the bulk of my attention on.

#### Mismatched elevation

```

Segment 1
( 4526.752318, 4139.720330, 300.0000 )
( 4469.434118, 3945.645154, 300.0000 )
Segment 2 (Ignored)
( 4526.752318, 4139.720330, 302.0000 )
( 4364.685250, 4187.885777, 301.0000 )
Point of Intersection
( 4526.752318, 4139.720330, 300.0000 )
( 4526.752318, 4139.720330, 302.0000 )

```

- 2) **Overlapping Segments** – I see this other type of “error” as more of a “notice” or an “alert” that I treat like a flag that is pointing out something. For me this is more often than not a lower level ‘error’ that very frequently has no impact in the final results of the overall surface model. More on this later.

#### Overlapping Segment

##### Segment 1

```
( 4364.685250, 4187.885777, 301.0000)
( 4247.547666, 4139.720330, 300.0000)
```

##### Segment 2 (Ignored)

```
( 4364.685250, 4187.885777, 301.0000)
( 4247.547666, 4139.720330, 300.0000)
```

I think the issue that has made this subject matter tricky to understand is the variations in terms and explanation that these ‘errors’ have been exposed to (at least for me).

Take the **InRoads Help** description for example, where it says:

*“You may choose whether to check for crossing breaklines or mismatched elevations by turning on the Extended Data Checks option. This verification process checks to see if any breaklines cross, if breaklines cross at mismatched elevations, or if breaklines overlap.”*

Here is where it starts to get muddy.

The **Help** appears to describe three issues:

- 1) If any breaklines cross
- 2) If breaklines cross at mismatched elevations
- 3) If breaklines overlap

First, you should notice that these are all ‘linear’ issues, but I’ll revisit that later (because what about “point” conflicts?)

Not to get too nitpicky (but I am ; ) notice the wording “*check for crossing breaklines or mismatched elevations*”. But again per the **Help**, one of the “checks” is to see “*if breaklines cross at mismatched elevations*”.

Okay, so are there really three problems? Crossings, mismatches and overlaps?

The InRoads **Error Log** reports on only two very specifically named problems:

- 1) **Mismatched Elevation**
- 2) **Overlapping Segment**

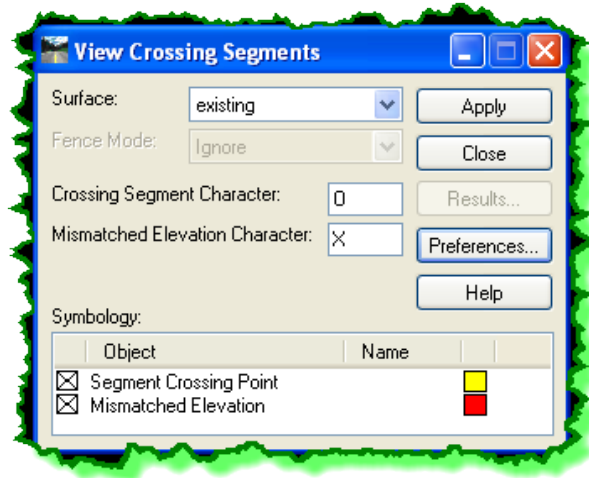
And neither of these mentions anything about a ‘crossing’. However if you refer back to how the report presents the problem areas you’ll see that the **Mismatched Elevation** is showing a **Point of Intersection**. This is the “crossing” that is being talked about. So, from this extrapolation, you should equate a **Mismatched Elevation** as an issue with a “crossing”.

#### VIEW CROSSING SEGMENTS:

At the risk of losing you I’m going to jump to the **View Crossing Segments** command because of the similarities regarding what I’m talking about.

This tool was very briefly mentioned earlier, and we’ll delve a bit deeper into it right now and relate it back to the **Triangulate Surface** command *Error Log* report.

First, notice the name of the command – **View Crossing Segments**. Here we go again with the ‘crossings’ ;) ...but does this command do more than just show ‘crossings’?



As mentioned, this tool places a symbol where ‘errors’ occur in the surface model. You can choose whatever character you want but an “X” and an “O” are pretty commonplace.

Because I’m such a fanatic about terms, let’s look at the terminology used here.

Midway on the dialog box there are fields to enter the character for the:

- **Crossing Segment**
- **Mismatched Elevation**

At the bottom, the **Symbology** is defined for the:

- **Segment Crossing Point**
- **Mismatched Elevation**

When this command is run, the **Results** button becomes active and allows you to view the “Error Log”, which happens to be the same report from the **Triangulate Surface** command identifying:

- **Mismatched Elevation**
- **Overlapping Segment**

Let’s take one thing at a time.

It’s pretty obvious that the term “**Mismatched Elevation**” is strung throughout the software. But what exactly is a “*mismatched elevation*”? Looking at the report from the **Triangulate Surface** tool, showing the “intersection”, it looks like a ‘crossing’ problem, remember?

The **Help** for the **View Crossing Segments** command says: “*This command displays a symbol at the point where two breaklines cross and another shape where crossing breaklines have different elevations at a common x,y coordinate.*”

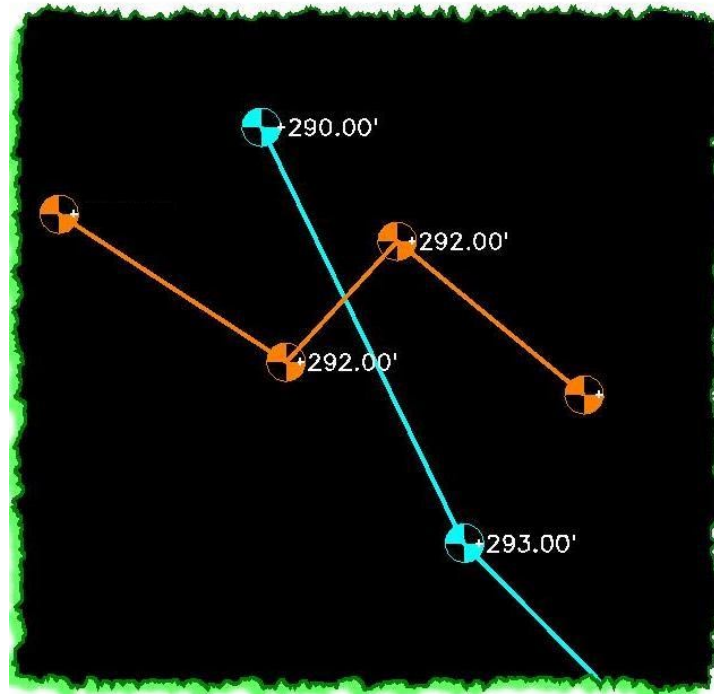
Hmmm... interesting ... but still muddy.

It goes on to define:

- **Segment-Crossing** - ... *the point where breaklines cross.*
- **Mismatched-Elevation** - ...*where the crossing breaklines have different elevations.*

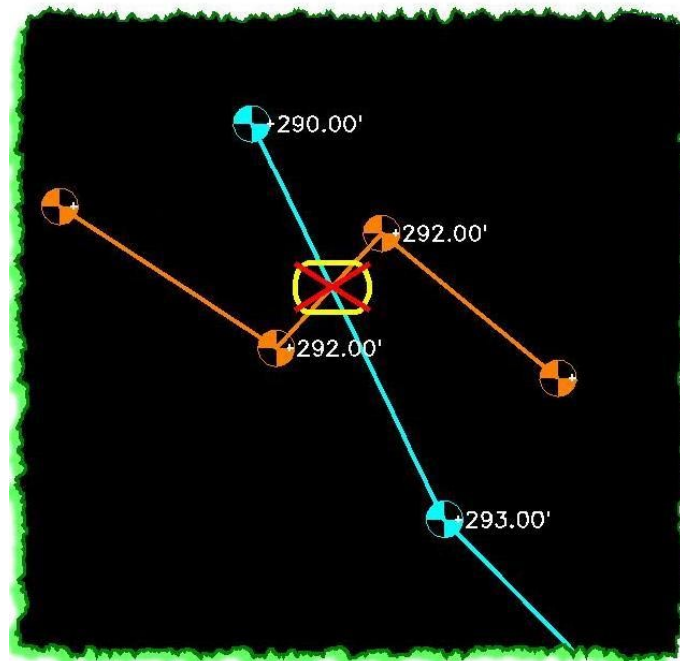
So, let’s go with that ...a **Segment Crossing** is a location where two breaklines cross. End of story.

And they may or may not have a different elevation at the intersection location. In a slightly broader sense it's the location where two linear items in a surface come into contact, from a planimetric perspective. (Remember the much earlier discussion about the surface model being a "planimetric" model.)

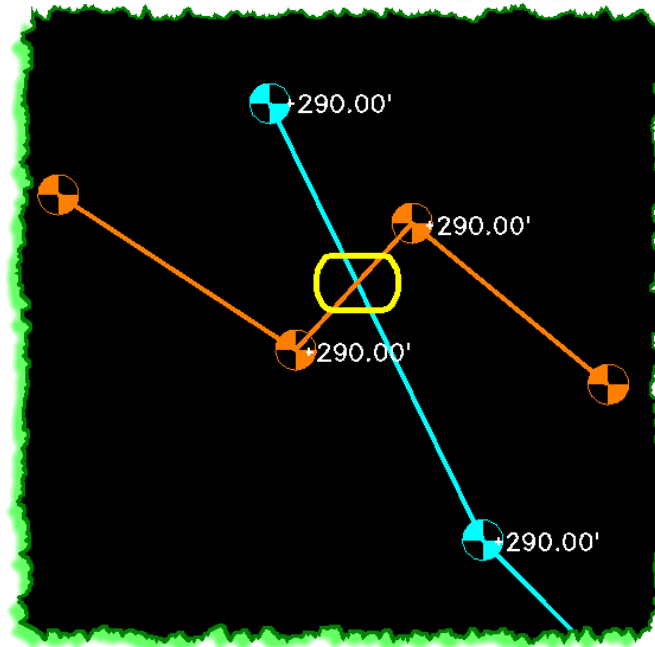


A **Mismatched Elevation** is a location where two breaklines cross, and two different elevations are derived at that location from each breakline segment at the intersection location.

In the example shown here, if the elevations are conflicting the **View Crossing Segment** command display will look like this:

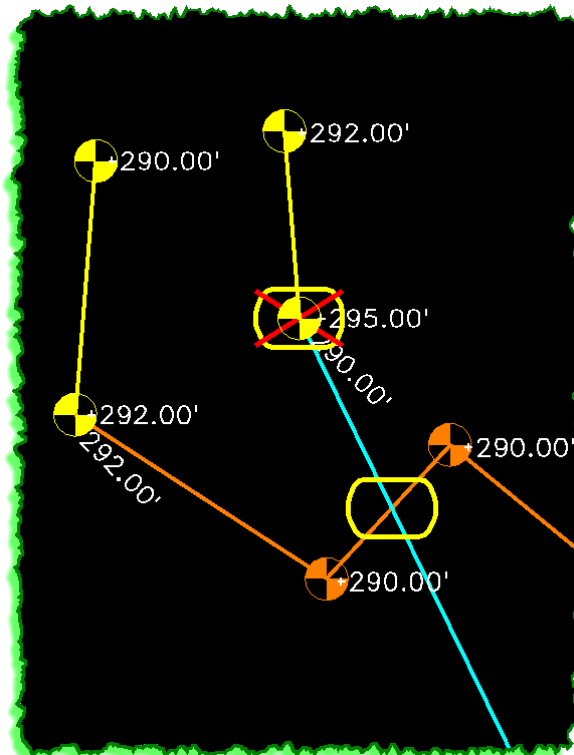


If the elevations are not conflicting it will show like this:



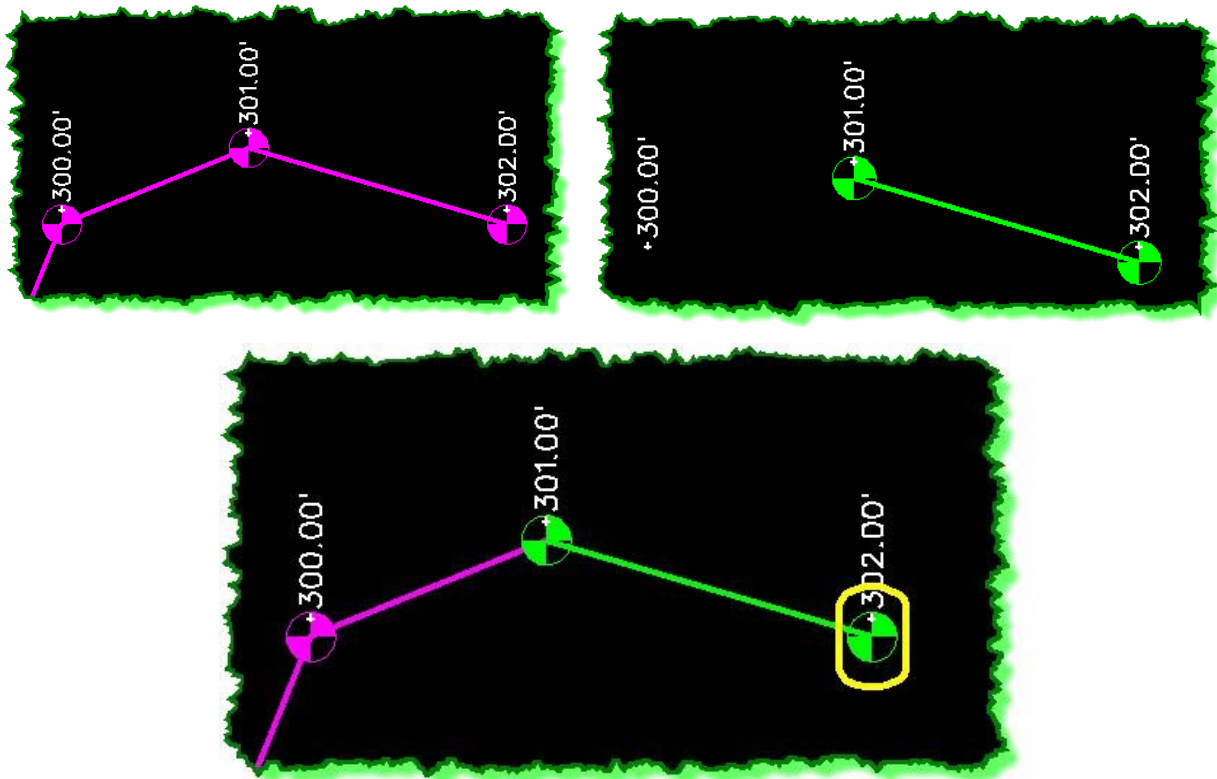
Regarding a **Segment Crossing** display, it's required that the two breaklines touch within the extents of the line, either between the vertices or somehow 'along' it in order for that character to be displayed. If two breaklines are simply touching at their 'ends' then that won't show as a **Segment Crossing** as illustrated below where the left yellow breakline is touching the orange breakline.

But if those breaklines contain an elevational difference, as shown by the right yellow breakline as it's touching the cyan breakline, then the symbols shown should display in the CAD file.

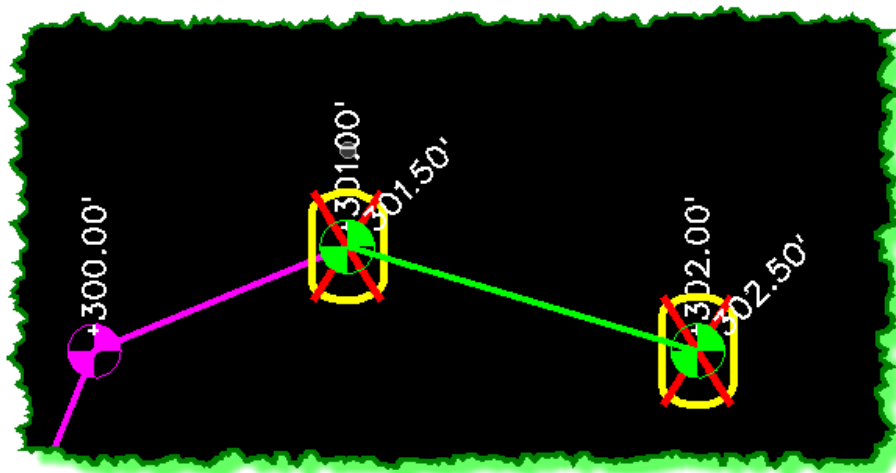


**Overlapping Segments** while identified in the **Results**, are not specifically categorized on the command dialog box itself, but are only partially displayed by this command. These **Overlapping Segments** are only shown by a single **Crossing Segment Character** on the end of the overlapping segment, not on both sides.

The picture shown below consists of a purple breakline, with a green breakline on top of the last segment of the purple breakline. The last picture shows how the **View Crossing Segment** command identifies this occurrence by placing a **Crossing Segment Character** on the last vertex of the overlapping segment. (The elevations are identical in this example.)



However, if the elevations are not the same on the overlapping segment then this command captures that issue differently and places both characters on both sides of the overlapping breakline like shown here.



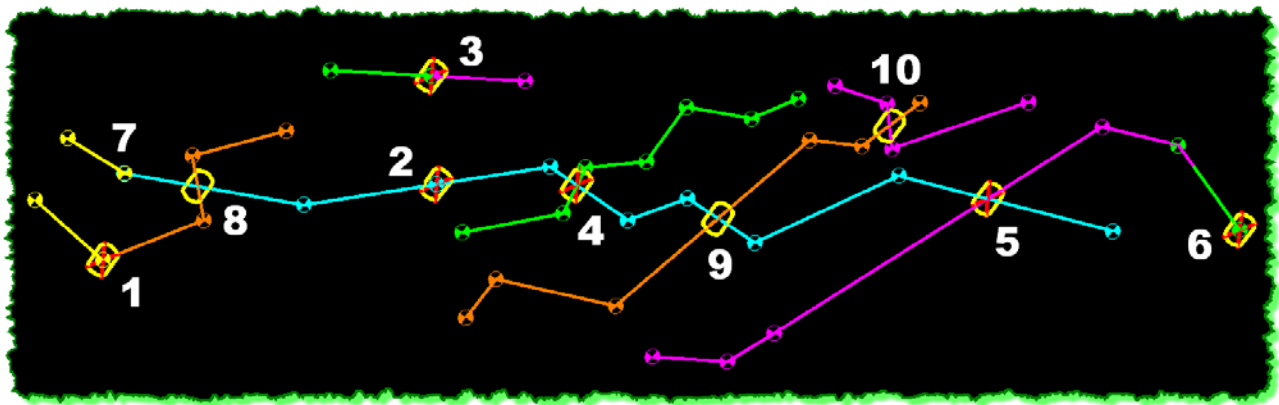
I hope you are all somewhat tracking with these explanations. But now for a bit of bad news... sometimes the elevational conflicts on breaklines that are simply touching each other at the ends don't show up using this tool. I haven't been able to identify exactly why, but can confirm that the display of the mismatches and crossings doesn't always show when some breaklines end where another breakline starts, regardless of the fact that there is an elevational bust involved.

The thing to understand here is that just because the **View Crossing Segments** command doesn't place a character on the issue, doesn't mean that the triangulation isn't doing the same thing as it always does ... "ignoring one of the offending elevations / segments". You just don't know it and won't see it listed in the report.

What that means is that you need to make sure that you are very attentive when working with your data. You really need to be on your toes all the time. When you are creating contours, looking at triangles, creating profiles and sections ... you need to be in constant 'alert' mode for anything that appears even slightly odd or unexpected. Don't move forward when you see anything strange ... hunt it down and find out what's going on because it could be one of these errors that has failed to expose itself.

### ERROR LOG ANALYSIS:

This is a portion of some testing that I was doing for this technical topic and consists of various breaklines strategically placed to induce problems and check the reaction of the software against various surface construction scenarios. (It's basically just a bunch of junk thrown into a DTM to see how the software reacts to the mess I've made ;) ... but masterminded!!!



Here is the **View Crossing Segments Error Log Report** from my testing surface and my notes just to give you a better perspective on what you can expect. I added this so you can begin to correlate the *Error Log* with the CAD display with the surface data itself. (What you don't see in the following report are those items that were valid issues yet didn't appear in the report. And "no", InRoads won't number the report and correlate it with a corresponding number in the CAD file. I wish.)



**1) Mismatched elevation****[These breaklines were touching end to end with different elevations]**

Touch 2 (Ignored)

( 2919.748, 3043.134, 290.000)

( 3066.328, 3031.170, 293.000)

break6

( 3066.328, 3031.170, 292.000)

( 3161.128, 3176.024, 290.000)

Point of Intersection

( 3066.328, 3031.170, 293.000)

( 3066.328, 3031.170, 292.000)

**2) Overlapping Segment****[These breaklines were also touching end to end with different elevations]**

break1

( 3277.276, 3290.097, 290.000)

( 3430.583, 3440.722, 293.500)

break7 (Ignored)

( 3430.583, 3440.722, 293.000)

( 3562.359, 3570.193, 295.000)

**3) Overlapping Segment****[These breaklines were also touching end to end with different elevations]**

Touch 4

( 3181.364, 3484.535, 293.000)

( 3319.181, 3572.584, 295.000)

Touch 5 (Ignored)

( 3319.181, 3572.584, 293.000)

( 3447.287, 3654.429, 293.000)

**4) Mismatched elevation****[These are crossing breaklines with an elevational intersection issue]**

break7

( 3562.359, 3570.193, 295.000)

( 3715.860, 3575.145, 296.000)

break2 (Ignored)

( 3624.254, 3523.153, 295.500)

( 3609.400, 3602.379, 295.500)

Point of Intersection

( 3615.115, 3571.895, 295.344)

( 3615.115, 3571.895, 295.500)

**5) Mismatched elevation****[These are crossing breaklines with an elevational intersection issue]**

break7 (Ignored)

( 4027.813, 3887.099, 299.000)

( 4362.049, 4018.317, 300.000)

break4

( 4247.548, 4139.720, 300.000)

( 4016.482, 3571.368, 299.000)

Point of Intersection

( 4167.070, 3941.770, 299.417)

( 4167.070, 3941.770, 299.652)

**6) Overlapping Segment****[These were overlapping breaklines with an elevational issue on one side]**

Overlap 1

( 4364.685, 4187.886, 301.000)

( 4526.752, 4139.720, 302.500)

break4 (Ignored)

( 4526.752, 4139.720, 302.000)

( 4364.685, 4187.886, 301.000)



Other notable items are:

- Number 7 has two breaklines touching end to end with conflicting elevations; however nothing was displayed in the CAD file, or in the *Error Log* report ...an Inexplicable anomaly.
- Numbers 8, 9 and 10 are **Crossing Segments** that do not contain any elevational issues at their intersection. These are displayed in the CAD file but are not contained in the *Error Log* report.

## RESOLVE CROSSING SEGMENTS:

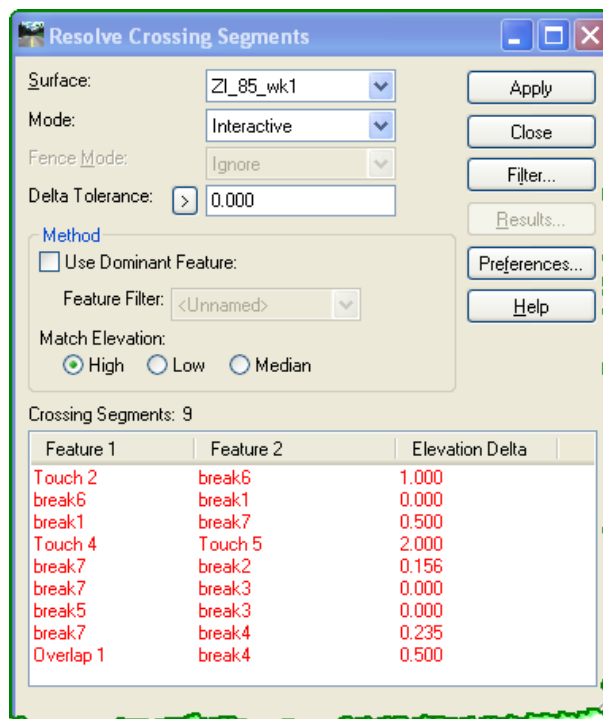
In light of all of this new information, let's take a look at the **Surface > Utilities > Resolve Crossing Segments** command and how it relates to the identification of these 'errors' in the surface.

The operation of this command was very briefly discussed earlier and only a bit more orientation will be added here. (You'll have to attend one of my classes for the full scoop on this tool ; )

First, I personally do not use the **Mode** set to **Automatic** except in very select cases. Sweeping, wholesale surface editing of errors are a bit too aggressive and cavalier in my opinion ...but I've recently been able to face the fact that I'm "old school" in some places and this new-fangled auto-patching of problems is just a bit too much for me to digest ; )

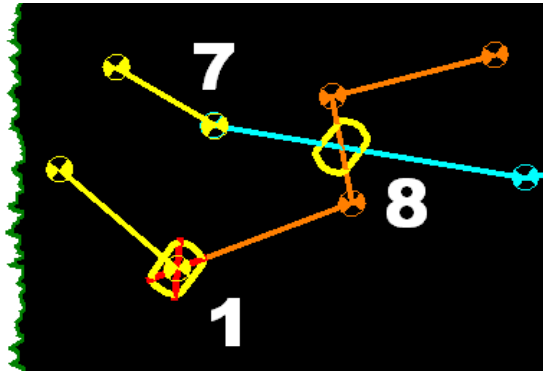
After setting the **Mode** to **Interactive**, the **Delta Tolerance** to **> 0.00**, and clicking **Apply**, the following **Crossing Segment** results are shown in the lower list window. (This is using the same testing surface that I've been using for this technical tip so we can compare it to items from our earlier discussion.)

After applying this command the lower list window feeds back whatever it found in the surface. As you can see, it's found everything that was initially listed in the earlier *Error Log*.



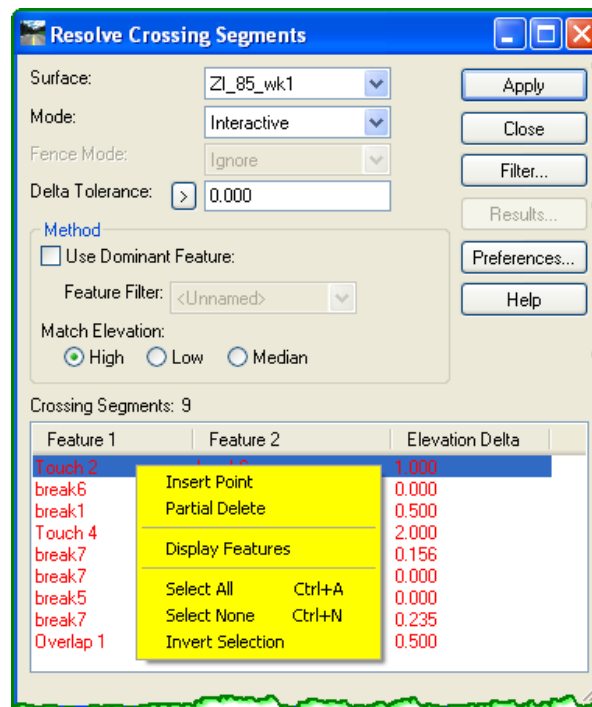
In addition to the *Error Log* entries, it also lists the 3 crossing breaklines that did not have any elevational conflict at their intersections. You can see these with the **Elevation Delta** of **0.000**. These were displayed in the CAD file with the **Crossing Segment Character** but were not listed in the report since they didn't pose any real problem in the surface model.

As a side note, that one error that failed to show up (#7) in both the **Triangulate Surface** report and the **View Crossing Segments** command has also failed to show up here.



Lesson = keep your eyes wide open!

This tool is very handy in that when you select an issue from the lower list window it will move your MicroStation view to center on that location (at your current zoom level) and highlight the problem location. You can also right-click on one of the **Crossing Segment** pairs in the list window at the bottom to expose a shortcut menu. This menu is intended to help you with some display and resolution options, but I don't use it very much at all except for the occasional **Display Features** option.



One notable item on this command is the capability to sort the **Crossing Segments** in the lower list window. I often sort them by **Feature 1** or **Feature 2** or **Elevation Delta** to fine-tune my analysis of the surface and focus on certain features or issues.

## SUMMARY OF TERMS:

Let me summarize and encapsulate the relevant terms so far:

- **Crossing Segments** – This shows up on the **View Crossing Segments** tool and refers to that location in the surface model (DTM) where two breakline segments form an intersection. This location may or may not contain an elevational conflict. Specifically this terminology is used on that dialog box to identify a specific **Character** to place at these locations when that tool is run. Consider this location a high-level look at whether or not your linear surface data crosses over other surface data.

This term is also used on the **Resolve Crossing Segments** tool, and has a similar usage as a broad brush-stroke term that means any and all locations where two linear surface features collide planimetrically.

- **Segment Crossing Point** – This also shows up on the **View Crossing Segments** tool and is referred to in the **Symbology** portion of that dialog box. Specifically it is used in reference to setting up the CAD display for viewing the location of any **Crossing Segments** as described above.
- **Mismatched Elevations** – This term shows up in the surface **Error Log** coming from either the **Triangulate Surface** command or the **View Crossing Segments** tool. It is also noted on the **View Crossing Segments** dialog box several places. Every time this is used it is intended to refer to any location in the surface model where two linear features intersect with a resulting difference in elevation. **Mismatched Elevations** are the most significant thing to locate and resolve when constructing a DTM.

During the Triangulation process InRoads must address these programmatically if they are encountered in order for a surface to be generated. The *Error Log* will indicate how InRoads resolved the encounter with any **Mismatched Elevations** by noting that one of the segments has been “*ignored*”. InRoads, programmatically selects one of the segments in order to complete the triangle formation process. Taking these ‘automated’ repairs at face value is a dangerous practice, and the user has to eventually become skilled at consciously addressing these problems.

- **Overlapping Segment** – More often than not this term is used to identify when a linear surface feature is positioned directly on top of another linear feature. These two features may or may not have conflicting 3D information. Most all roadway surface models contain this class of data since the **Exterior Boundary** of the surface almost always traces directly over the daylight cut / fill breaklines. This is very common and not an issue. The *Error Log* will report that one of these ‘duplicates’ is “*ignored*” for triangulation purposes which is inconsequential in the final surface construction. These only need to be reviewed and corrected if they contain conflicting elevational information.
- **Crossing Breaklines** – This is really just a generic term referring to any instance where linear data in a surface model is crossing over other linear data. Sometimes within the software this shows up when the linear data is merely touching each other at either the beginning or end, but either way, it doesn’t necessarily indicate that a problem exists. With that said, this term is usually used with a concerned tone to indicate that a problem exists and the surface is in need of verification.

OMG!!! ... That's it!! ... I'm done!!

... For now ; )

Coming next ... the big finale - ZI Issue 86: **The Final Word on DTM Conflicts!!**