VTrans Intersection Tools

Tools related to updating Nodes & NodeLegs, and related attributes in rdsmall & rtlogpts

UpdateRoadAttributesAddNode (AfterSplitCreateNode) ![icon]

Select the two new rdsmall features created by splitting an existing rdsmall feature

Duplicates the functionality of the original UpdateRoadAttributes Button (from HMS_Toolbar), but also creates a Node feature at the split location, determines the next available NodeID from the rdsmall fields StartNodeID and EndNodeID, and populates the Node and both rdsmall features with that ID. It is generally easier to use the original UpdateRoadAttributes button separately, and to then use the AddAll tool in the Intersection Toolbar to add all intersection features together.

AddLegsOnly (LegsOnly)

Select the Node feature and any intersecting rdsmall features that do not have corresponding NodeLegs

Creates NodeLegs for a selected Node from the selected intersecting rdsmall features. Can be used to create any number of legs (one, all, or some) associated with that Node, depending on the number of intersecting rdsmall arcs that are selected. If a single rdsmall feature is selected, the selected Node feature **must** intersect the correct end of that rdsmall feature. NOTE: The NodeLegCount of the Node feature and the new NodeLeg features will be updated to reflect ONLY the number of selected rdsmall features. If there are additional pre-existing NodeLegs associated with that Node feature, then the NodeLegCount will either have to be updated manually, or the entire "intersection" can be "updated" with the UpdateNodeAndLegs (UpdateAll) Button.

AddNodeAndLegs (AddAll)

Select all rdsmall features that intersect at the location where a Node and NodeLegs are needed. No other features in any layers may be selected for this to work.

Creates Node and NodeLegs features for the selected rdsmall features that intersect at a single location, assigning unique NodeID and NodeLegIDs to each feature, and updates each rdsmall feature to reflect the NodeID at the appropriate end of the rdsmall arc (Start or End)

UpdateNodeAndLegs (UpdateAll)

Select a Node feature and all intersecting rdsmall and NodeLeg features

Updates the geometries of the selected Node and NodeLegs features to reflect the geometry of the selected rdsmall features. This tool assumes that attributes correctly associate each NodeLeg with its 'parent' rdsmall feature (Currently uses StartNodeID and EndNodeID that each leg 'remembers' or 'learns' from its 'parent' rdsmall feature, but I'm considering changing it to FAID). The number of selected NodeLegs MUST match the number of selected rdsmall features. (Tool will fail otherwise).

CheckRelationalAttributes

Select one or more rdsmall features

Evaluates all intersection-related features that are spatially associated with each selected rdsmall feature and pops up messages indicating missing features (except rtlogpts) or incorrect attribution (in rdsmall and NodeLegs features, and rtlogpts too) pertaining to how these features relate to each other. Assumes rdsmall.FAID, Nodes.NodeID, NodeLegs.NodeLegID, rtlogpts.POINTID are correct and unique, and that

topology is intact (with the exception of a flipped NodeLeg, because one end still matches end of rdsmall feature, but an alert message will show).

UpdateRelationalAttributes

Select one or more rdsmall features

Same as CheckRelationalAttributes, but updates the attributes of all features to be consistent with the current topology/geometry and rdsmall.FAID, Nodes.NodeID, and NodeLegs.NodeLegID values updates values to be consistent with the existing topology instead of popping up messages. Message alerts regarding missing features can be coded to still appear (or not). Does not correct for flipped NodeLegs, but updates all attributes as if leg were digitized correctly (e.g. StartEnd)

NOTE: Generally, the other Intersection tools will ensure everything is 'all set' with the Node, NodeLeg, and rdsmall features after a split or unsplit. However, the NodeLegs at the OTHER end of the rdsmall features created by splitting will be unaware of the change of FAID and of the identity of its neighboring Node.

UpdateRoadwayAttributes

Select one or more rdsmall features

Same as UpdateRelationalAttributes, but updates different fields of the NodeLegs (and maybe eventually Nodes) features, notably values that can be obtained from rdsmall such as TWN_LR, RTNUMBER, RDFLNAME, SpeedLimit, etc. A Null value in rdsmall will not, however, overwrite a non-null value in the intersection data, and a warning message pops up instead. This is to prevent data collected for the intersection data being deleted. Because of the prevalence of Nulls in rdsmall.SpeedLimit, there is no pop up warning for a for that particular scenario, and the NodeLeg value is not updated.

UpdateNodeLegCount

Select all features associated with a Node (Node, NodeLegs, and rdsmall arcs)

Uses same input selection as the UpdateNodeAndLegs button, but does not update geometry. Instead it updates the same attributes as the UpdateRoadwayAttributes button, with the addition of NodeLegCount. Currently this tool updates different fields of the NodeLegs (and maybe Nodes) features, notably values that can be obtained from rdsmall such as TWN_LR, RTNUMBER, RDFLNAME, SpeedLimit, etc. A Null value in rdsmall will not, however, overwrite a non-null value in the intersection data, and a warning message pops up instead. This is to prevent data collected for the intersection data being deleted. Because of the prevalence of Nulls in rdsmall.SpeedLimit, there is no pop up warning for a for that particular scenario and the NodeLeg value is not updated.

UpdateNode (to be developed, maybe)

Select the Node feature and the appropriate number of rdsmall features to indicate the Node's new location (at least two intersecting arcs, except select only one arc if the Node is a 'Dangle')

Updates the geometry of the selected Node feature to reflect the geometry of the selected rdsmall feature(s). The only advantage over manual snapping is the assurance that the Node is snapping to a rdsmall feature (not another feature in another feature class), and ensures that the selected arcs defining the Nodes location intersect at a single point (except for dangle scenarios, in which case the tool requires that only one end of the selected arc is a dangle.

NewNodeID (to be developed, maybe)

Select the Node, its NodeLegs, and intersecting rdsmall features

NewNodeLegID (to be developed, maybe)

Select one or more NodeLegs

Applying Intersection tools to typical workflows

Split Option 1 (pseudonode and legs)
1. Split rdsmall arc using Esri's Split Tool
2. Select both new arcs, and run [Arc ID icon] on the *HMS Toolbar* (same procedure as before intersection data)
3. Select both new arcs and run **AddAll**
4. Run **UpdateRelationalAttributes** (with the two arcs still selected)

Split Option 2 (pseudonode and legs)
5. Split rdsmall arc using Esri's Split Tool
6. Select both new arcs, and press the [Arc node icon] on the *HMS Toolbar*
7. Select both new arcs and the new Node feature, and run **LegsOnly**
8. Run **UpdateRelationalAttributes** (with the two arcs still selected)

Moving a Simple Intersection Workflow
Before use:
1. Relational attributes must be intact/valid before editing road centerline geometries (and updating corresponding intersection features). Best way to ensure this is to select the rdsmall arcs and run **CheckRelationalAttributes,** and then **UpdateRelationalAttributes (**if necessary) before editing an intersection.
2. Note that splitting rdsmall arcs (and updating FAID, adding new Node and NodeLegs, and new Start/End Nodes, etc) changes the relational attributes for neighboring nodes, so do not split arcs while in the middle of moving intersection features. Finish all moves and update everything before commencing with splits.
Moving the intersection:
3. Temporarily ignoring the Node and NodeLegs, simply edit the rdsmall arcs as per the rdsmall manual, ensuring that their endpoints all snap to the same location after they have been moved.
4. Select the intersecting rdsmall arcs, the Node, and the NodeLegs (that are still reflect the previous rdsmall alignment) and run **UpdateAl**l.
After use:
5. It's good practice to then select all rdsmall arcs that may have been updated and run **CheckRelationalAttributes**, though for a change that only involves moving rdsmall arcs, everything should be set.
6. If an rdsmall arc needs to be flipped, flip it manually, and run **UpdateRelationalAttributes** before proceeding. After everything is done, the NodeLegs on the other end of the flipped arc will also need to be flipped manually.

Add Intersecting Road Workflow
Option 1: (preferred) Adding new Node only after splitting/updating existing arc AND adding/updating new intersecting arc(s)
1. Split rdsmall arc using Esri's Split Tool
2. [Arc ID icon] Update road attributes (original tool that updates ArcID and FAID, etc, but doesn't add node)
3. Paste new arc
4. [Arc ID icon] Select new arc and run tool to update ArcID and FAID for a single arc.
5. Edit new arc manually so one end vertex intersects the other two arcs at their split point
6. Select all 3 rdsmall arcs (no other features in any layers!), run **AddAll**
7. If the other end of pasted arc is a dangle:
   a. Select new pasted arc only, run **AddAll** again to create dangle Node and NodeLeg
8. If the other end of pasted arc needs to connect to an existing node:
   a. Manually edit pasted arc so that the end vertex intersects other existing Node
   b. Select only the pasted arc and the existing Node it now intersects

      c. Run **LegsOnly** to create a leg for this arc

      d. Make sure the Node and all its NodeLegs (including the new NodeLeg) all know that NodeLegCount has changed

9. Select each rdsmall arc intersecting any of the new Nodes (or all at once), and run **UpdateRelationalAttributes.** This will update NextNodeID values in NodeLegs to include the new Node.

10. Double-check by running **CheckRelationalAttributes** again, and also **UpdateRoadwayAttributes** for good measure (as long as you are sure that the SurfaceType in rdsmall is correct, not the value(s) in NodeLegs that have been edited by RPC_Editor.

Option 2: Adding new Node after Split before adding intersecting road

1. Follow the Split Workflow (pseudonode and legs) above

2. Add the intersecting arc, and edit it manually ensuring that it intersects the existing arcs at the same location

3. Select the Node and the new arc, and run **LegsOnly** to create a leg for the new arc that intersects the new Node (either now or later need to manually update NodeLegCount for Node and NodeLegs intersecting this Node)

4. Select only the newest arc and run **AddAll** to add the Node and NodeLeg to the other end of the new arc

5. Run **UpdateRelationalAttributes** with each/all of the affected arcs selected

Applying Intersection tools to typical workflows (in progress)

Workflow for flipping rdsmall arcs: (If these steps are done out of order, no problem, just make sure to execute #4 and #5 last)

1. Flip rdsmall arc
2. Adjust OneWay value of rdsmall arc if necessary (rare, only for one-way segments)
3. Flip the NodeLegs on both ends of the arc (I can eventually make a tool that will do this in one step, but it's easy to do manually)
4. Select the rdsmall arc and run **UpdateRelationalAttributes** to update NextNodeID, and Start/End values for each NodeLeg.
5. While still selected, select "CheckRelationalAttributes" to make sure they were also updated, and run "UpdateRelationalAttributes" if necessary

Workflow after other rdsmall (geometry only) changes that affect Node or NodeLeg geometry:

1. Select all rdsmall arcs, NodeLegs, and Nodes and run **UpdateAll**
2. Run **CheckRelationalAttributes** to make sure neighboring Nodes and NodeLegs weren't impacted (though best to run **CheckRelationalAttributes** BEFORE editing rdsmall geometry, because **UpdateAll** doesn't work unless relational attributes were – and remain – intact.

Workflow after updating any rdsmall attributes that are relevant to MIRE data

1. With any/all updated rdsmall segments selected (it doesn't matter if Nodes or NodeLegs are also selected) run **UpdateRoadwayAttributes**

Updating an intersection where "Two Legs point towards the same Node" (Quasi-loop Nodes):

(See summary of instructions with images in the next section of this guide)

- Until the tool automatically recognizes this scenario and updates using a different method, the user must update these intersections in two steps, pretending that one of the approaches (corresponding rdsmall_arc and NodeLeg) does not exist each time.

  1. Instead of executing **UpdateAll** with the Node and ALL of its intersecting rdsmall arcs and NodeLegs selected, one of the rdsmall_arcs that "reconnect" with each other at a neighboring Node (and that arc's corresponding NodeLeg) must remain unselected. Run **UpdateAll**.
  2. Repeat the above process, but leaving the OTHER rdsmall_arc/NodeLeg pair unselected, and run **UpdateAll** again.
  3. Manually update NodeLegCount value on Node/NodeLeg features to reflect the true number of NodeLegs intersecting the Node. This "work-around" makes the Node think there is one NodeLeg less than there actually is, due to the number of NodeLegs selected each time the tool is run.

Updating "Loop" Node intersections:

(See summary of instructions with images in the next section of this guide)

1. This process also needs to be done in two steps, but instead of unselecting the looped arc (all arcs remain selected), the vertex at one end of the looped arc needs to be moved so that it no longer intersects the Node. The NodeLeg corresponding to the same end of the looped arc as the moved vertex must also be

unselected. Then run **UpdateAll**, which will see the formerly looped arc as only connecting to its Node once.

2. Replace the "detached" vertex of the loop arc to its original position.
3. Repeat step 1, but move the vertex at the other end of the looped arc, and unselect its corresponding NodeLeg before running **UpdateAll**.
4. Replace the "detached" vertex.
5. Manually update NodeLegCount to reflect all the NodeLegs intersecting that Node (two of which correspond to the looped arc).

Note: **UpdateAll** also triggers the Editor Tracking date of all selected features, even if the updated feature is no different than the original.

Applying Intersection tools to Loop Nodes and Quasi-Loop Nodes

UpdateAll for a loop Node:
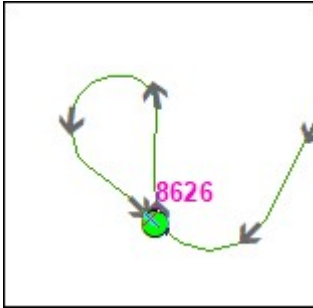
1. Unsnap one end of the rdsmall loop arc
   a. Select Node, both rdsmall arcs, and two NodeLegs that are Not associated with the unsnapped end of rdsmall (left image below)
   b. Run UpdateAll
2. Unsnap other end of the rdsmall loop arc
   a. Select Node, both rdsmall arcs, and two NodeLegs that are Not associated with the unsnapped end of rdsmall (right image below)
   b. Run UpdateAll
3. Manually set Nodes.NodeLegCount = 3 (NodeLegs.NodeLegCount isn't currently updated with this tool)
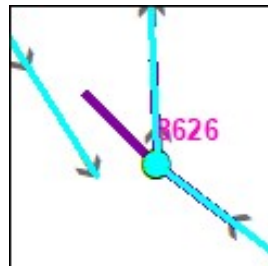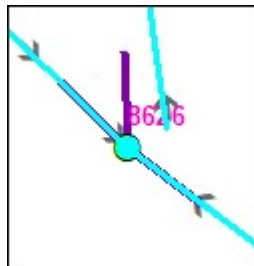
UpdateAll for a quasi-loop Node: (Left image below – Two NodeLegs 'point' towards the same neighboring Node)
1. Select Node, two rdsmall arcs (one of which does not point to the same neighboring Node, and only one that does), and the two corresponding NodeLegs of the selected rdsmall arcs. (Top-right image below)
   a. Run UpdateAll
2. Select Node, two rdsmall arcs (this time selecting the other arc that leads to the same neighboring node), and the two corresponding NodeLegs of the selected rdsmall arcs. (Bottom-right image below)
   a. Run UpdateAll
3. Manually set Nodes.NodeLegCount = 3 (NodeLegs.NodeLegCount isn't currently updated with this tool)
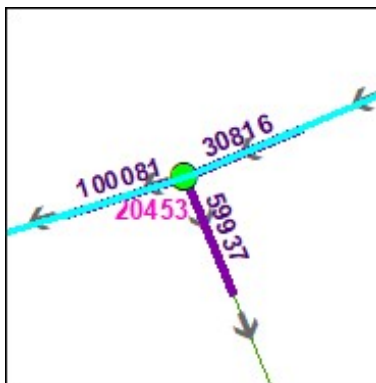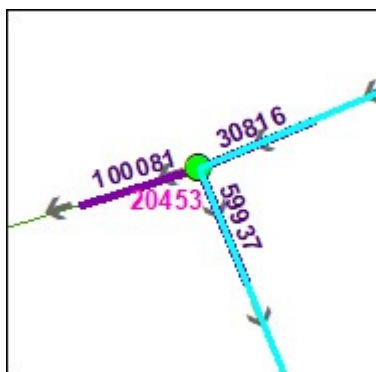
Tools with Node vs rdsmall perspectives

Tools with a rdsmall perspective:
The **CheckRelationalAttributes**, **UpdateRelationalAttributes**, **CheckRoadwayAttributes** and **UpdateRoadwayAttributes** tools all work from the perspective of one or more selected rdsmall arcs. For each selected rdsmall arc, only its two Nodes and two NodeLegs (determined geometrically so they don't have to be selected) are considered and/or updated. These "rdsmall-perspective" tools do not, however, consider NodeLegs of unselected rdsmall arcs, even if other NodeLegs intersecting the same Node are being considered (because THEIR rdsmall arc IS selected). For this reason, these tools are unable to geometrically determine or update the NodeLegCount of their Nodes and NodeLegs.

These rdsmall-perspective tools were developed for two reasons: First to more efficiently transfer manually updated rdsmall attributes to NodeLegs (assuming geometry is correct). Secondly, they ensure that any updates to rdsmall that affect FAID, StartNodeID, or EndNodeID values (e.g. splitting an arc and adding a new Node) are pushed to the NodeLegs at the other end of the affected rdsmall arcs (i.e. the NodeLegs of neighboring Nodes along the connecting rdsmall arc). More specifically, when rdsmall is split the resulting arcs have new FAID values. Although the tools used to create a new Node and its NodeLegs took into account the new FAID values, the pre-existing legs on the other ends of those new arcs will still "remember" the previous FAID value of the unsplit arc. UpdateRelationalAttributes will fix this.

Tools with a Node perspective:
The remaining intersection edit tools all relate to a one Node and its intersecting rdsmall and NodeLeg features. The Node can represent a dangle, pseudonode, or true node. Node-perspective tools consider existing selected features, make updates as necessary, and adds new features as necessary (depending on the tool used). **AddAll** requires that all the intersecting rdsmall arcs be selected (*but no other features can be selected in any other layers*), **LegsOnly** requires that the Node also be selected, and **UpdateAll** requires that the NodeLegs also be selected.
- If a Node needs to be created, the location is determined by the rdsmall selection set:
    - Multiple arcs must intersect at a single point, which defines the Node's location
    - A single rdsmall arc without a selected node is only relevant to the **AddAll** tool, and when a Node and NodeLeg need to be created at its "dangle end", so the other end must intersect at least one other rdsmall feature.
- If Node already exists, it must be selected when running a Node-perspective tool, and the selected Node's NodeID is pushed to all associated features. The **AddAll** tool will always create a new Node with a new NodeID, **LegsOnly** and **UpdateAll** require that the correct Node feature be selected.
- Some Node-perspective tools function for a subset of its approachs (rdsmall arcs and corresponding legs). It is possible to use **LegsOnly** if a Node and any number of its intersecting rdsmall arcs are selected. Likewise it is possible to use UpdateNodesAndLegs (**UpdateAll**) with a subset of the NodeLegs as long as the Node is selected and only the corresponding subset of rdsmall arcs are selected (i.e. equal number of rdsmall arcs and NodeLegs selected).
- **LegsOnly** for a single arc requires that the selected Node be snapped to the correct end of the rdsmall arc so that the tool knows which end needs the NodeLeg. (When multiple arcs are selected for **LegsOnly**, the ends that need NodeLegs are determined by identifying which end of each arc intersects the other selected arcs, so although the Node must be selected, the legs will be created even if the Node isn't snapped to that location).
- also require that the Node already be topologically in synch with its intersecting rdsmall arcs (and NodeLegs Some**UpdateNodeLegCount**, and. Although rdsmall can figure out which Nodes and NodeLegs are associated with it topologically, and "exchange" relational attributes with and "push"roadway attributes to those Nodes and NodeLegs, the rdsmall arc perspective doesn't investigate

Known Tool Limitations

1. **UpdateAll** doesn't work if two NodeLegs have the same NextNodeID. (See documentation above for work-arounds to update Loop Nodes and Quasi-loop Nodes).
2. None of the node-perspective tools to work properly on Nodes or NodeLegs that involve a looped rdsmall feature (FromPoint intersects ToPoint).

Intersection Data and Key Relationship Fields

rdsmall
- FAID: A unique identifier for each rdsmall feature
- StartNodeID: The NodeID identifying the Node feature snapped to the start of the rdsmall feature
- EndNodeID: The NodeID identifying the Node feature snapped to the end of the rdsmall feature

Nodes
- NodeID: A unique identifier for each Node feature

NodeLegs
- NodeLegID: A unique identifier for each NodeLeg feature
- NodeID: The NodeID of the Node feature that the NodeLeg belongs to
- FAID: The FAID value identifying the node leg's parent rdsmall feature
- StartEnd: Indicates whether the leg represents the Start of End of its parent rdsmall feature
- NextNodeID: The NodeID of the Node feature at the *other* end of the same parent rdsmall feature
- StartNodeID: Inherited or learned from parent rdsmall feature (helps determine NextNodeID)
- EndNodeID: Inherited or learned from parent rdsmall feature (helps determine NextNodeID)
- NodeID_NodeLegID_StartEnd: Join field that uniquely identifies every NodeLeg to a single end of a single rdsmall feature
- NodeID_TWNLR_JoinField: Join field that almost uniquely identifies the LRS measure for each NodeLeg (the only exceptions are for NodeIDs of looped routes. A Python dictionary keeps track of these Routes that have more than one valid measure where they intersect a particular Node. (Currently ~10 including some NFRs)

rtlogpts
- POINTID: A unique identifier for each rtlogpt feature
- NodeID: The NodeID of the Node feature at the same location as the rtlogpt feature

Topological Relationships

Nodes and rdsmall
- There exists one (and only one) Node feature with a shape identical to each rdsmall arc's FromPoint, indicated by rdsmall.StartNodeID
- There exists one (and only one) Node feature with a shape identical to each rdsmall arc's ToPoint, indicated by rdsmall.EndNodeID
- Each Node may correspond to the Start and/or End points of multiple rdsmall arcs.
- No two Nodes have the identical geometry

NodeLegs and rdsmall
- A 'Start' NodeLeg has a FromPoint that snaps to its parent rdsmall feature's FromPoint, and it also 'shares a segment' with its parent rdsmall feature.
- An 'End' NodeLeg has a ToPoint that snaps to its parent rdsmall feature's ToPoint, and it also 'shares a segment' with its parent rdsmall feature.

Nodes and NodeLegs
- A 'Start' NodeLeg has a FromPoint that snaps to its Node and its ToPoint should not snap to another Node, or intersect another NodeLeg
- An 'End' NodeLeg has a ToPoint that snaps to its Node, but its FromPoint should not snap to another Node, or intersect another NodeLeg
- When two adjacent Nodes are close together (snapped to the two ends of an rdsmall arc whose length is less than 10 meters) such that the default NodeLeg length (5 meters) would lead to overlapping NodeLegs, both nodelegs are truncated to ensure a small gap between them.

Attribute Relationships ('Relational Attributes')

Nodes and rdsmall
- rdsmall.StartNodeID and rdsmall.EndNodeID keep track of which Node feature is snapped to its FromPoint and ToPoint, respectively.

NodeLegs and rdsmall
- NodeLegs.FAID keeps track of the parent rdsmall feature's identity
- A 'Start' NodeLeg has NodeLeg.StartEnd = 'Start'
- An 'End' NodeLeg has NodeLeg.StartEnd = 'End'
- NodeLegs.StartNodeID and NodeLegs.EndNodeID are inherited (or 'learned') from the leg's parent rdsmall feature, as there is a possibility that FAID alone would change

Note: This attribute relationship is actually topological in nature, but is represented as attributes. To be truly attribute-related, rdsmall would require the fields rdsmall.StartNodeLegID and rdsmall.EndNodeLegID.

Nodes and NodeLegs
- NodeLeg.NodeID keeps track of its Node

Nodes or NodeLegs and rtlogpts
- rtlogpts.NodeID keeps track of which Node it should be snapped to.
- NodeLegs.POINTID could theoretically be used, but that would require additional maintenance. This relationship can be almost uniquely established via NodeID_TWNLR_JoinField (and is used to update the almost unique LRS measures for NodeLegs.

Relationship redundancy has its limits

NOTE: When edits are made that 'break' either the topological relationship between features or the attribute relationship between features, that relationship MUST be 'repaired' before making edits that 'break' the other type of relationship. **CheckRelationalAttributes** verifies proper topological relationships between features before verifying the relational attributes, so it is wise to run this tool before making any geometry changes. If Node or NodeLeg features have incorrect Relational Attributes information about their relationships with other specific features, their relationship to rdsmall and each other, either the topological relationship or the attribute relationship must be restored manually before a tool can be leveraged to update the other. Generally it is easier to manually repair the geometry, and use the **UpdateRelationalAttributes** tool to update the relational attributes, rather than the other way around. Then **UpdateAll** can be used to tidy up the manually updated NodeLegs. If UpdateAll is not run, some tools and QA/QC processes will notice that the NodeLeg isn't perfectly to "specifications" in that it isn't 5 meters long (or shorter when associated with rdsmall arcs < 10 meters long).

Intersection Toolbar VB.NET Shared Functions/Subs

AddNodeAndLegs.MakeLeg(pFeature_rds, newNodePoint, NewNodeID)

Update_IntersectionAttributes.MakeLeg(pFeature_rdsmall, pNode, NewNodeID, {StartEnd})

Update_IntersectionAttributes.Set_FAID_StartNodeID_EndNodeID(pFeature_rdsmall, pFeatureClass_Nodes, pFeatureClass_NodeLegs, pFeatureClass_rtlogpts, pMap, pWorkspace, pEditor, {roadwaytoo})

Update_NodeAndLegs.UpdateLegAttributes(StartEnd, NodeLegID, pFeature_leg, pFeature_rds, pEditor)

Check_IntersectionAttributes.CheckLegAttributes(StartEnd, NodeLegID, pFeature_NodeLegs, pFeature_rdsmall, pEditor)

Intersection Toolbar
AfterSplitCreateNode CheckRelationalAttributes CheckRoadwayAttributes

LegsOnly AddAll UpdateAll

UpdateRelationalAttributes UpdateRoadwayAttributes