Stream Toponym Matching

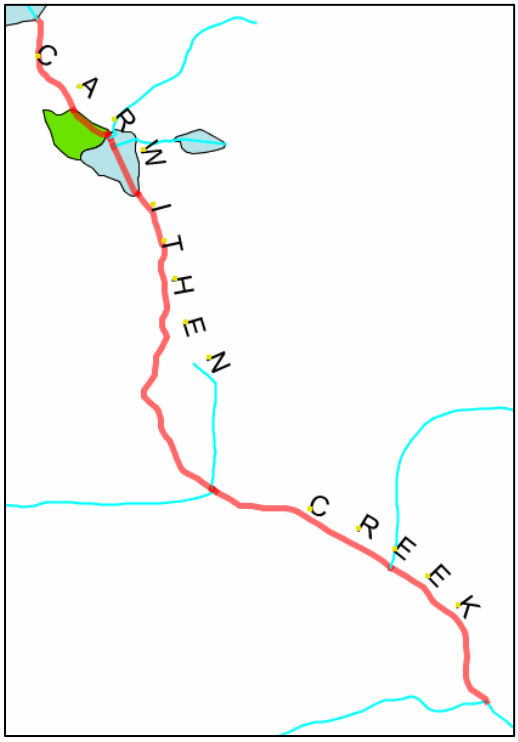
Submitted By: Justin Deoliveira, Programmer
Refractions Research Inc.
400 - 1207 Douglas Street
Victoria, BC, V8W-2E7
jdeolive@refractions.net
Phone: (250) 383-3022
Fax: (250) 383-2140

1 INTRODUCTION

A *Stream Toponym* is a collection of cartographic letters that are intended to name hydrographic features such as rivers, brooks, etc.

Currently there is no association between a stream and the toponym that names it. The only relationship between the two is derived from the spatial characteristics of the toponym: *A toponym should be very close to the stream it names.*

The following is an example of a toponym naming a stream. A stream is made up of a number of stream segments.

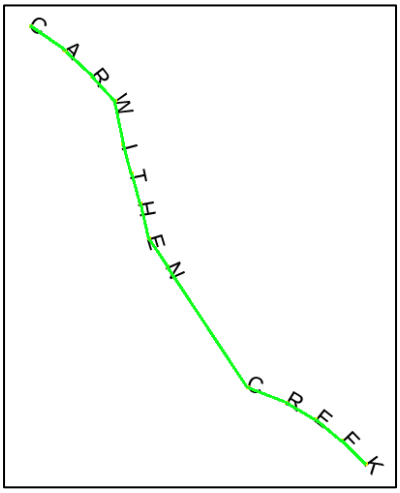


In the figure above we can see that the placement of the toponym “CARWITHEN CREEK” that it is intended to name the stream segments that are highlighted in red.

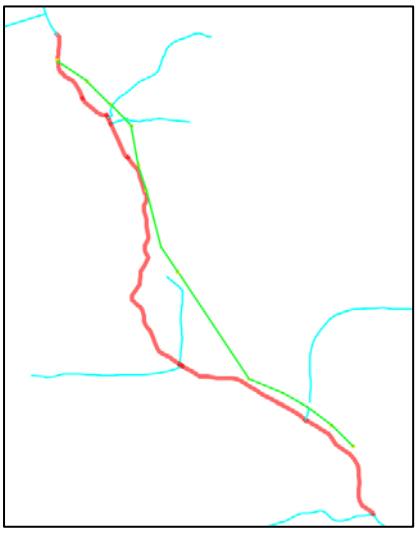
2 APPROACH

The stream toponym matching algorithm works on the assumption that *the baseline of the toponym should match geometrically well to the stream it is intended to name.*

The baseline of a toponym is obtained by connecting every point in the toponym with the point(s) adjacent to it. The green line in the figure below is the baseline.



If we overlay the baseline with the stream we see that the two follow a similar linear path which indicates a good geometric match.



3 THE ALGORITHM

The Stream Toponym Matching algorithm is the same algorithm used to conflate the 1:50k Watershed Atlas streams to the 1:20k Corporate Stream Network. See the 1:20k 1:50k Stream Matcher documentation for more details.

4 STREAM TOPONYM MATCHING UTILITY

The Stream Toponym Matcher is a utility that can be used with the Java Unified Mapping Platform (JUMP). When started, the utility performs some automated matching routines, and then presents the results to the user for manual verification and editing.

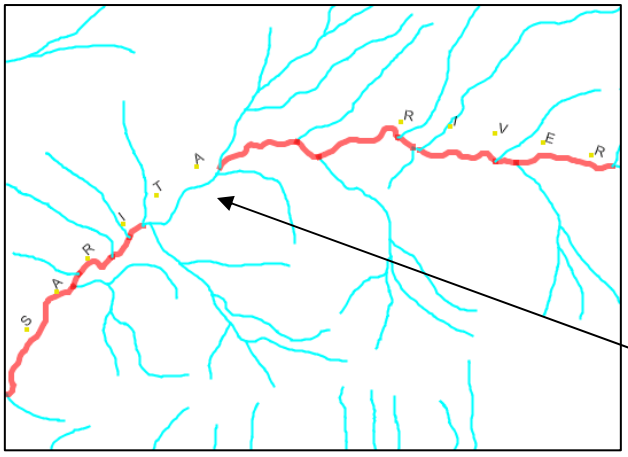
4.1 Manual Editing

During match editing, the following restrictions are placed on what can be a valid match:

1. All the stream segments matched to the current toponym **must** form a connected path.
2. All the stream segments matched to the current toponym **must not** form a forked path.

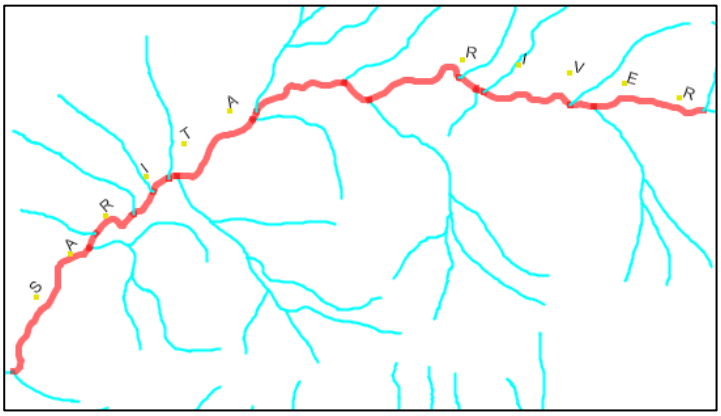
The following is an example in which the segments matched to the toponym do not form a connected path.

Invalid Match



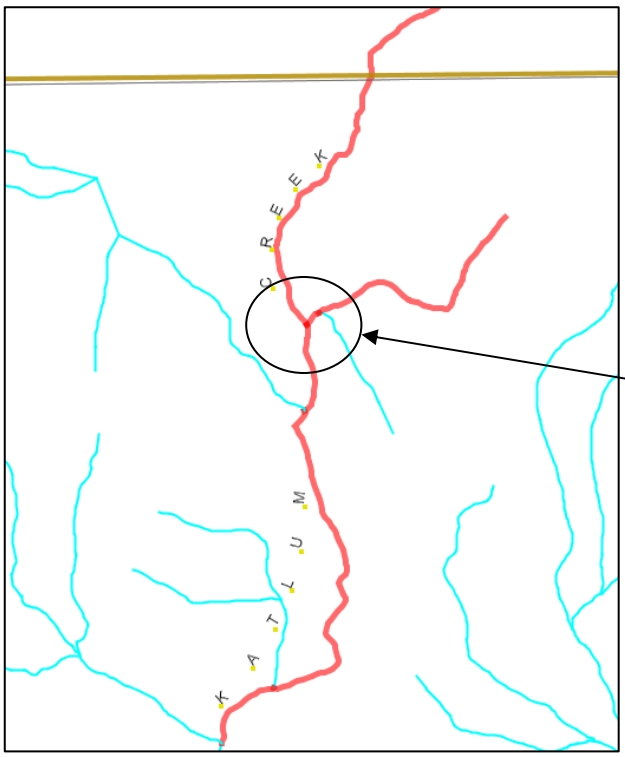
To fix this match the user would use the “Add” button to add the unmatched segment to the match. The result would be the following.

Valid Match



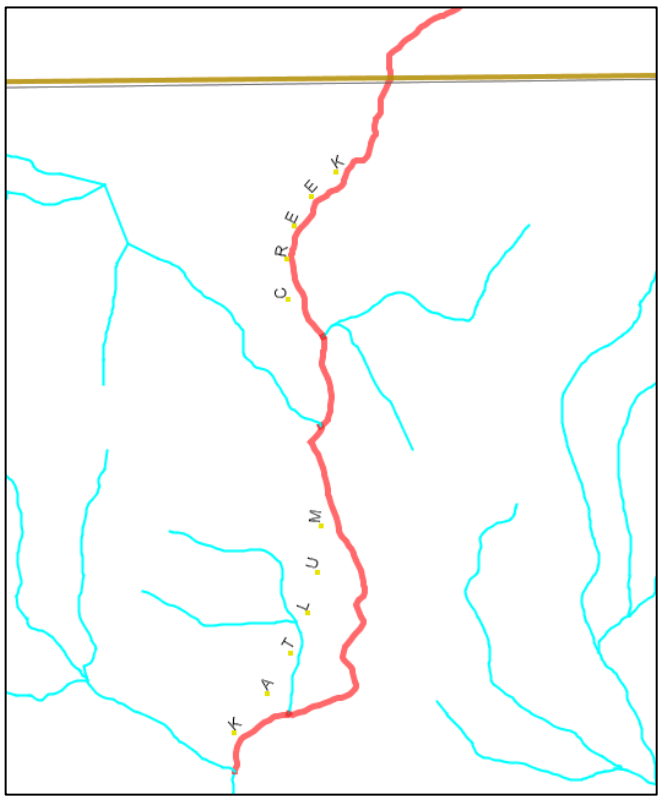
The following is an example of where the matched stream segments form a path that forks.

Invalid Match



To fix the match the user would use the “Remove” button to remove the segments that fork. The result would be the following.

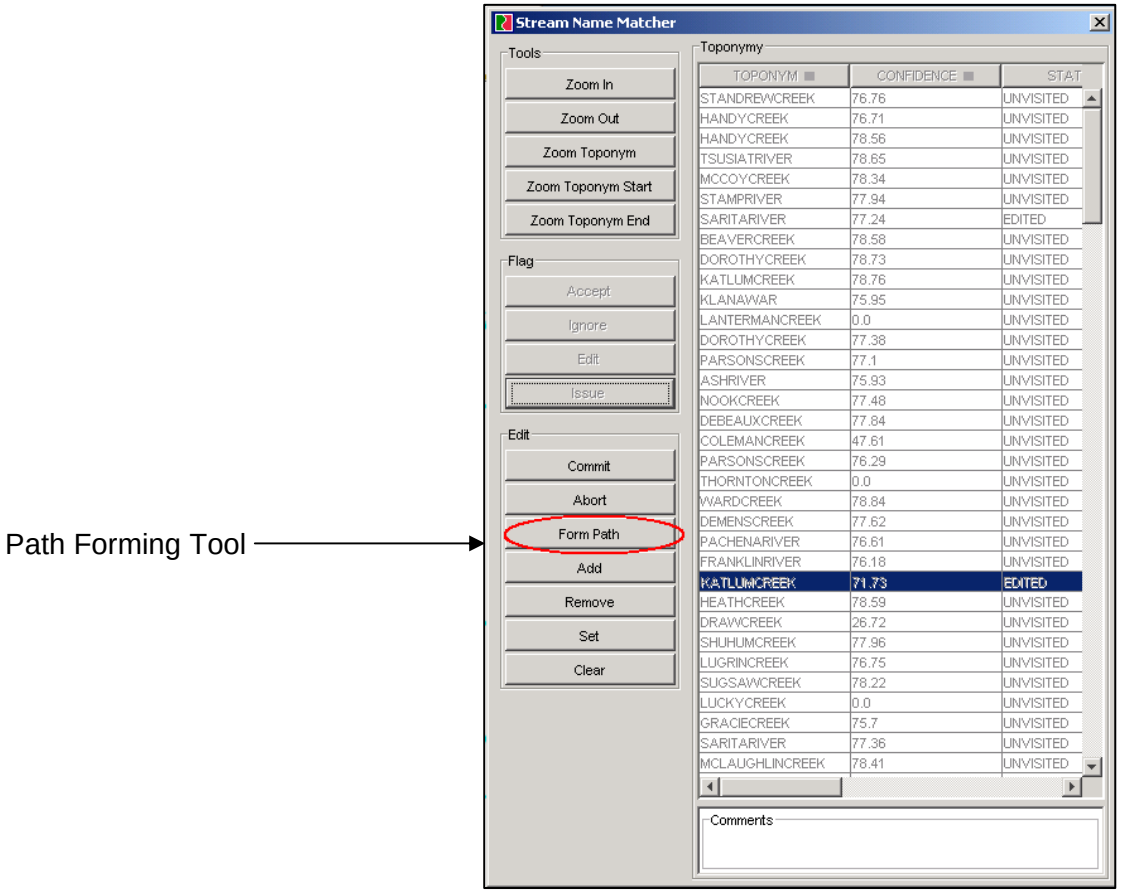
Valid Match



Note: The automated matching process will **never** produce any of the above cases. The only time they can occur is during the manual editing process.

4.2 Path Forming Tool

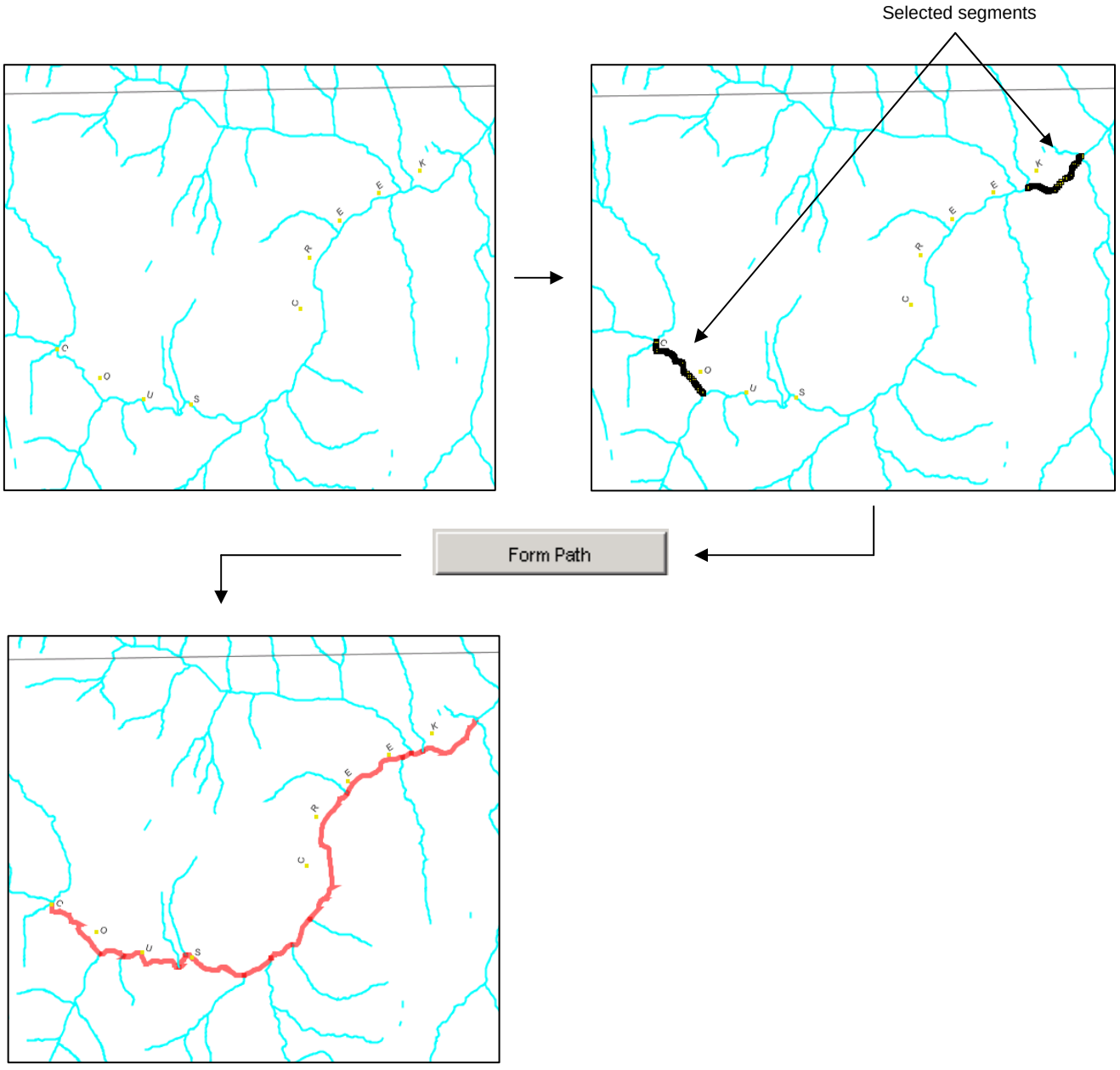
Most of the manual matches performed amount to making a connected path between two stream segments. The Path Forming Tool is a tool that allows users to do this easily. The Path Forming Tool is shown below and is only available in *toponym edit mode*.



The Path Forming Tool works as follows.

1. User selects two segments (using the Feature Selection Tool) and clicks the **Form Path** button.
2. The Path Forming Tool forms a path between the two selected segments.

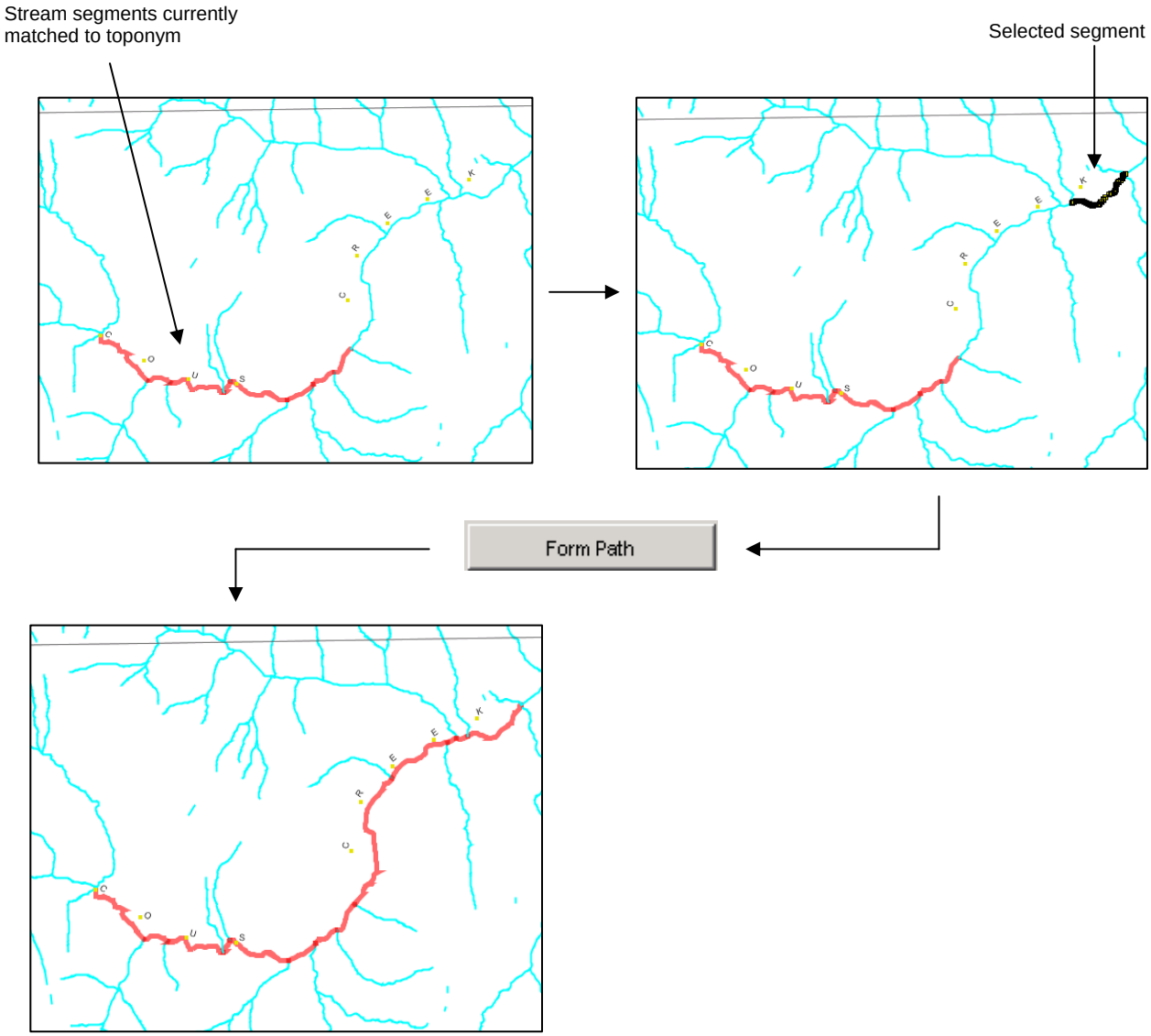
Below is an illustration.


Selected segments
Form Path

The Path Forming Tool can also be used as follows.

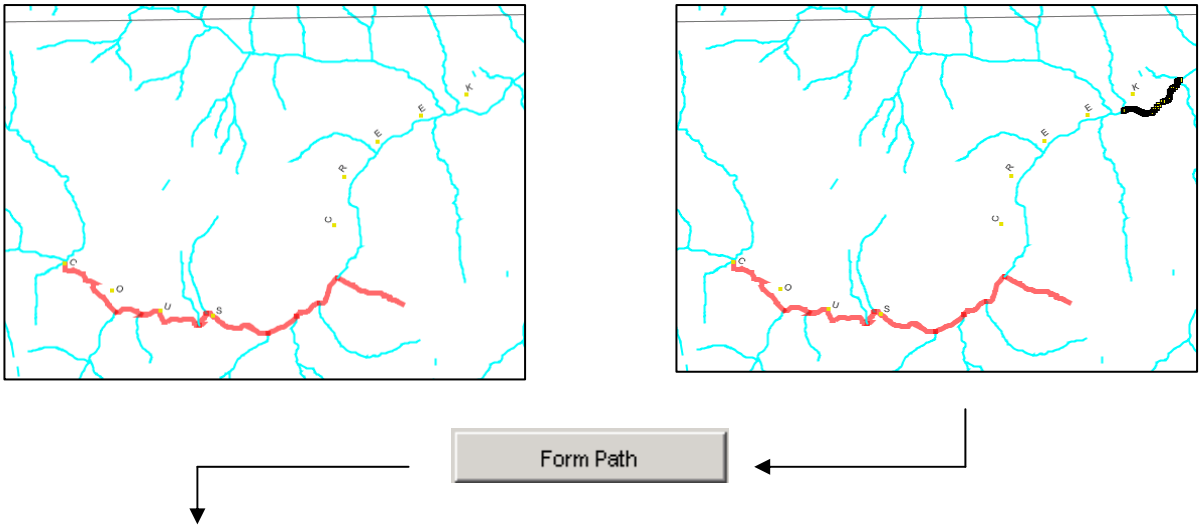
1. The user selects a **single** stream segment using the Feature Selection Tool and then clicks the **Form Path** button.
2. The Path Forming Tool forms a path between the selected segment and the current matched stream segments. Note: if there is no current match then the Path Forming Tool simply matches the selected segment.

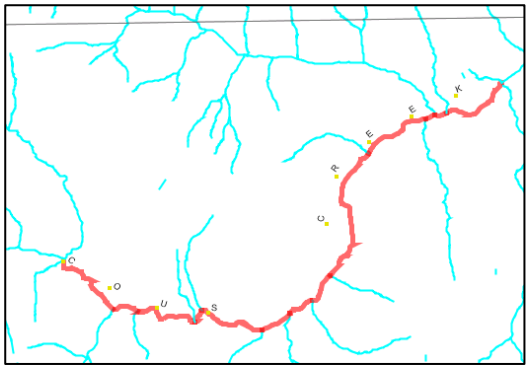
Below is an illustration.



Using the Path Forming Tool with a single selected segment will also prune of any segments that produce a fork.

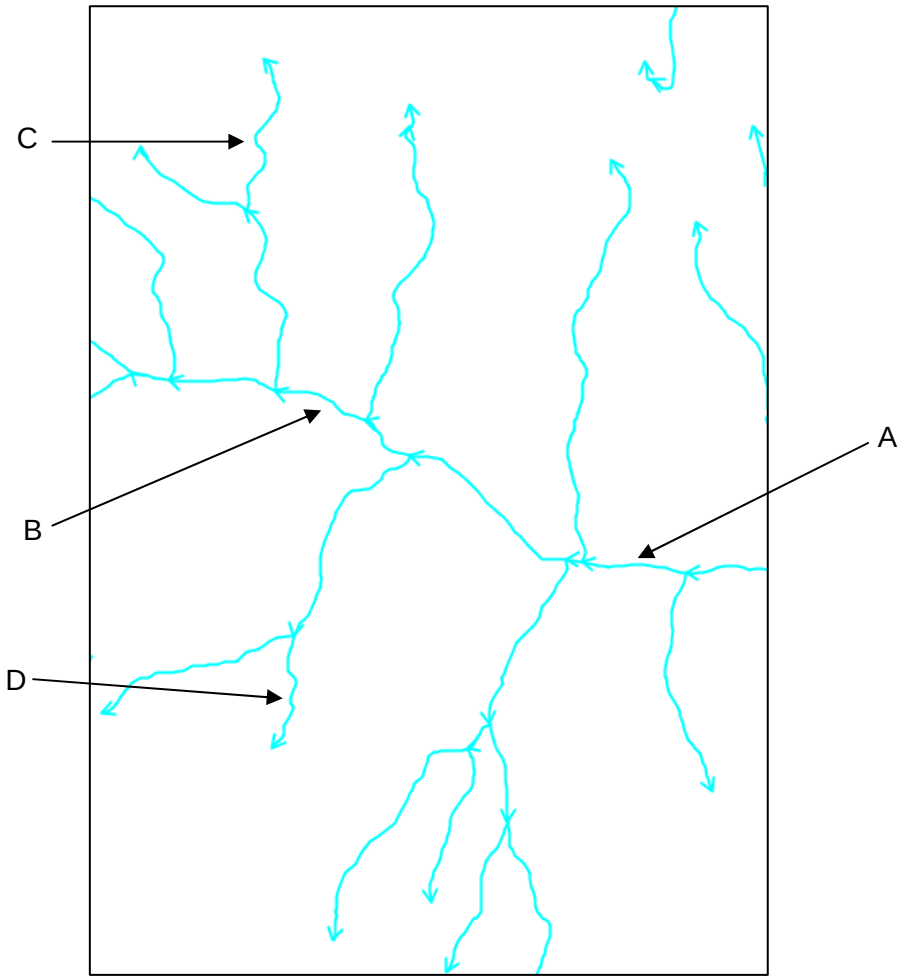
This is illustrated below.





Note: The Path Forming Tool can produce a path between two segments only if one of the segments is upstream of the other. In the event that the segments are not related in an upstream-downstream relationship, a path will not be produced. This is illustrated below.

Note: In the following diagram the blue arrowheads indicate upstream flow.



In the above diagram:

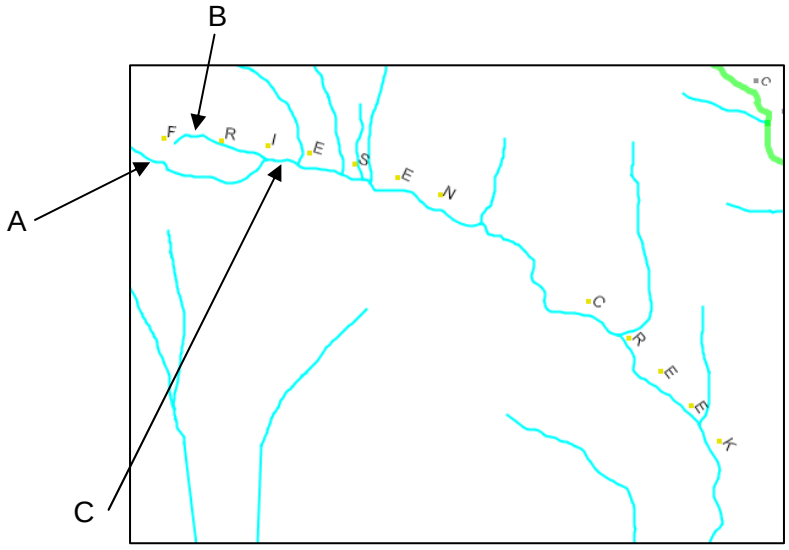
- B,C, and D are all upstream of A
- C is upstream B

Therefore the paths BD and CD can not be formed.

4.3 Manual Matching Recommendations

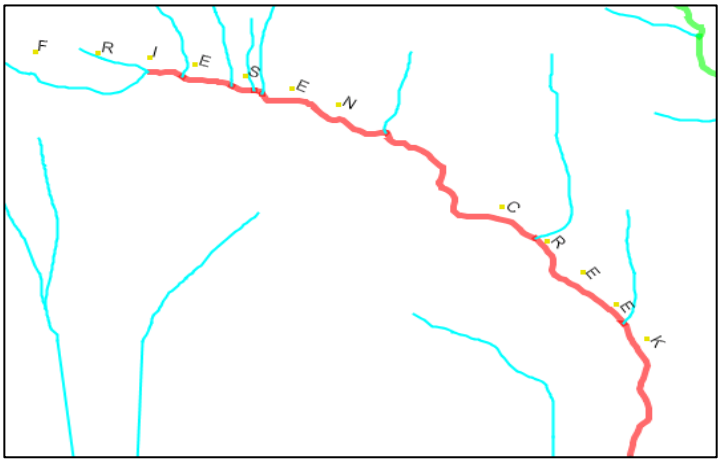
The following recommendations are made to the operator who is performing the manual matching.

1. Do not push ends of a match too far. If unsure shorten match as opposed to lengthening it.



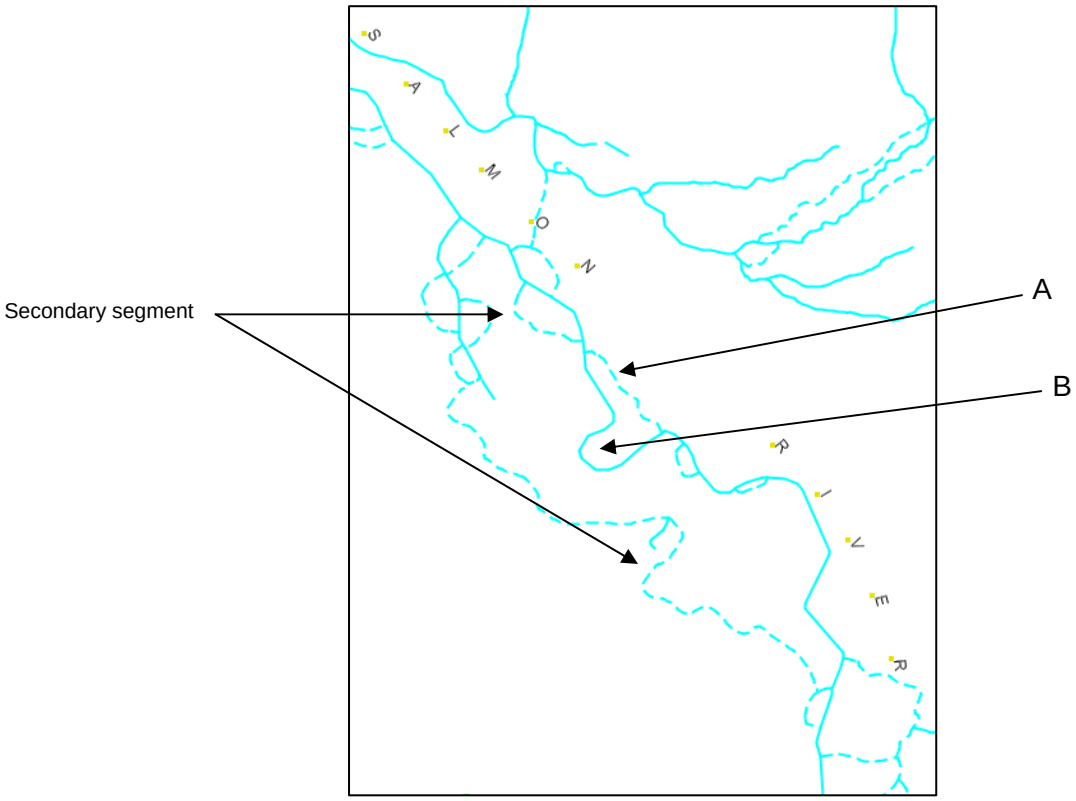
Consider the start of the toponym shown above. From the view we are given it is unsure whether or not “FRIESEN CREEK” ends at segment B or continues upstream and should be matched to segment A.

In this case the operator is encourage to stop the match at C as shown below.



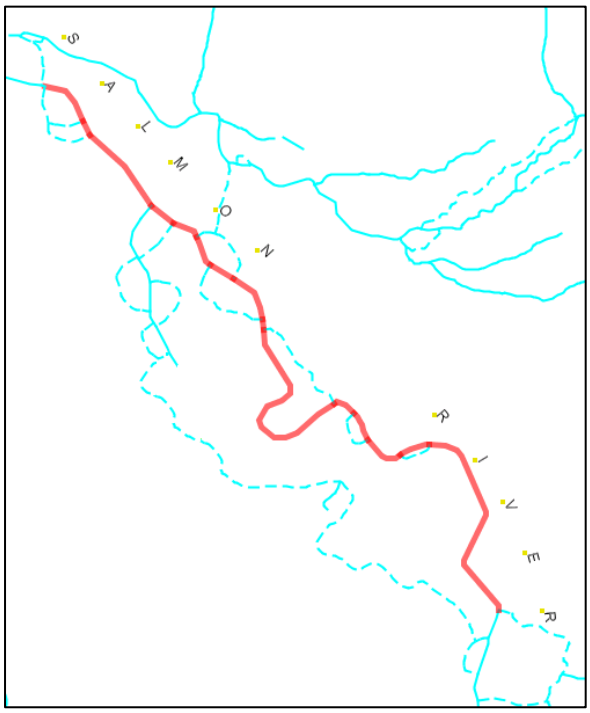
2. If possible, do not match to secondary segments.

Secondary segments represent secondary flows in the stream network.

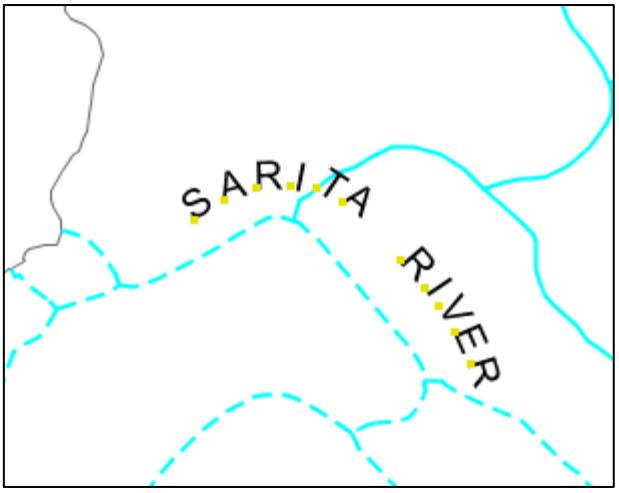


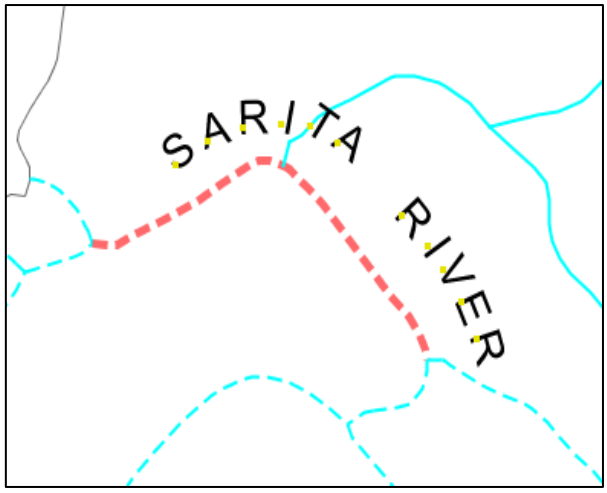
Consider the toponym shown above. Secondary segments are shown with dashed lines. In this case it is ambiguous as to what segments the toponym should be matched to. Should segment A or segment B be considered part of “SALMON RIVER”.

Whenever there is a choice between a primary segment and a secondary segment, the operator is encouraged to always choose the primary segment. Matching to a secondary segment is legal, but results in a warning message. The following would be the recommended match

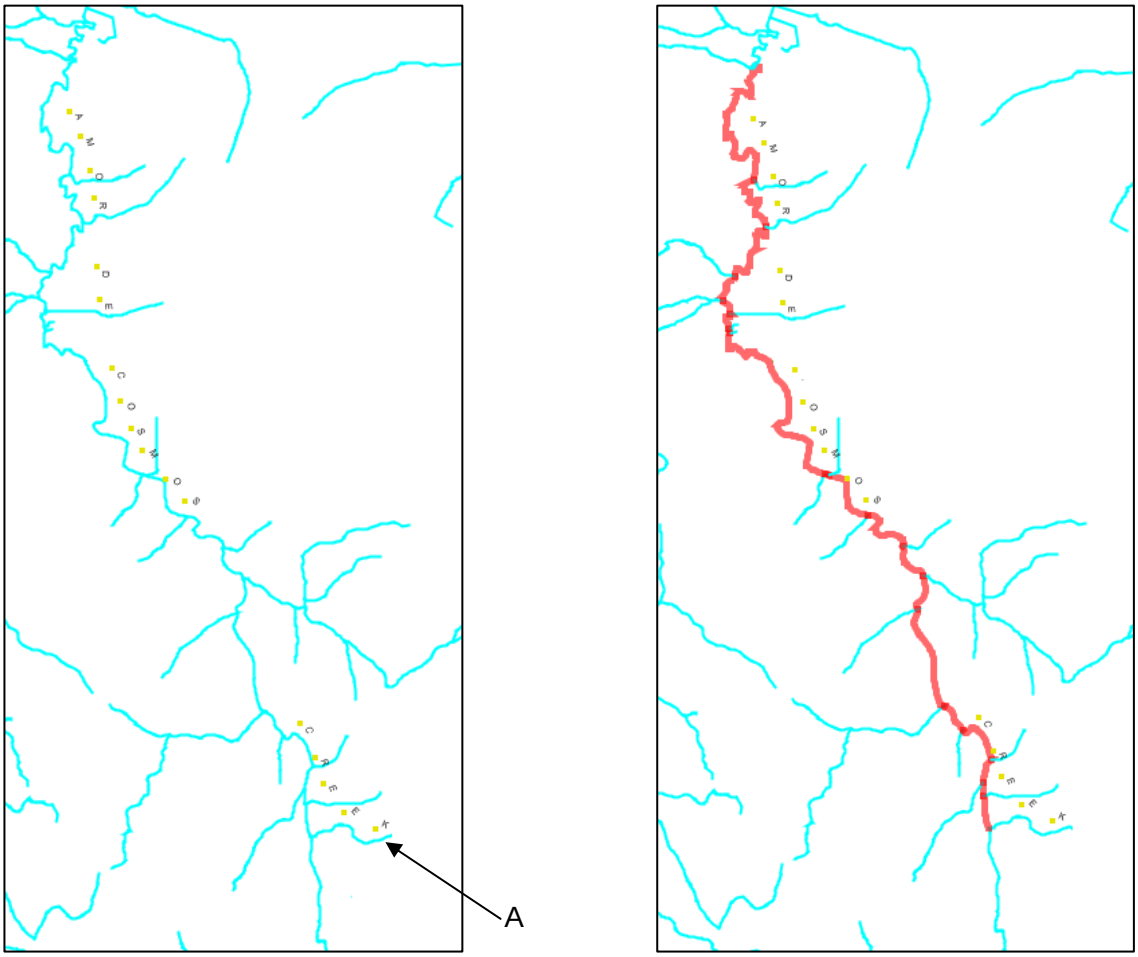


However there are cases when the only choice is to match to secondary edge, in this case it is okay to do so.





3. Do not match a large toponym to small attributaries.



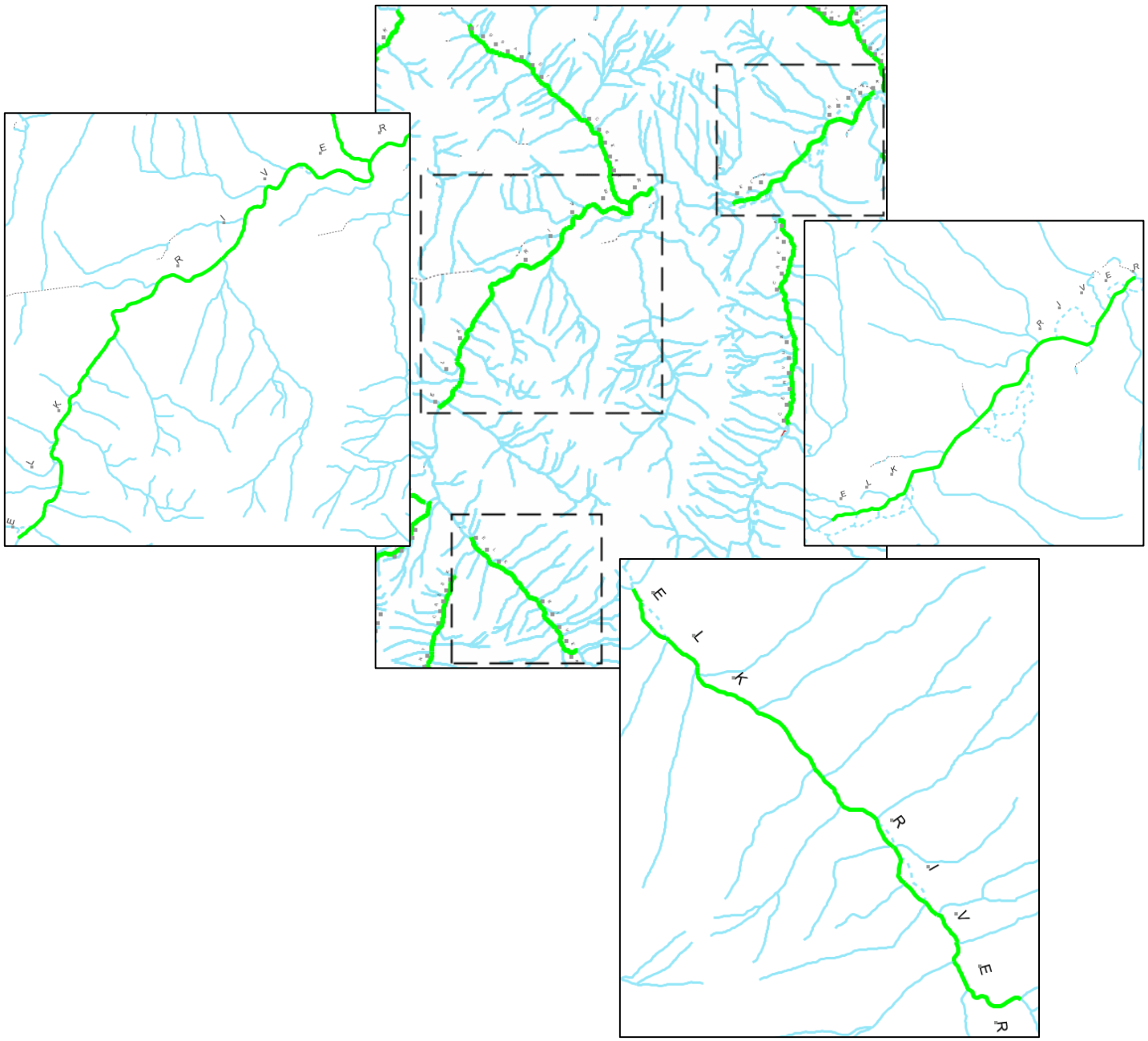
In the figure above on the left, the toponym seems to follow up the attributary labeled A. However this toponym is relatively large which means that the stream it names will be relatively large. For this reason it is unlikely that this small attributary will be part of the stream.

The desired match is shown on the right.

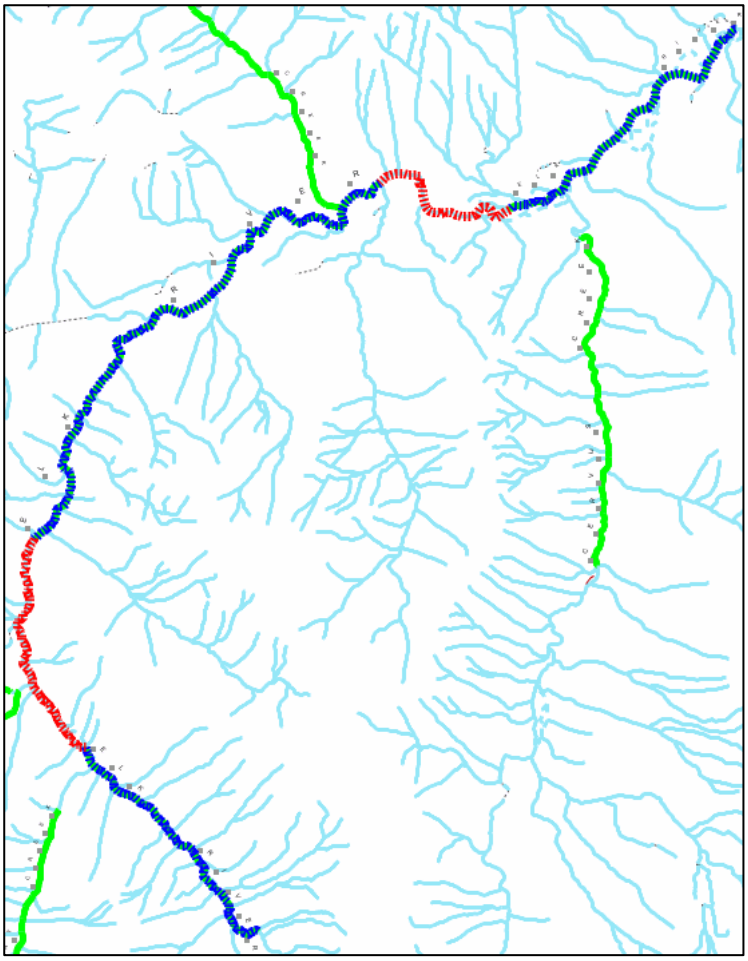
5 QA/QC TOOLS

5.1 Snake Generation

The result of the stream toponym matching session is a collection of named segments scattered throughout the stream system. This collection of named segments can be partitioned into disjoint sets in which two segments are in the same set if they are named to the same toponym.



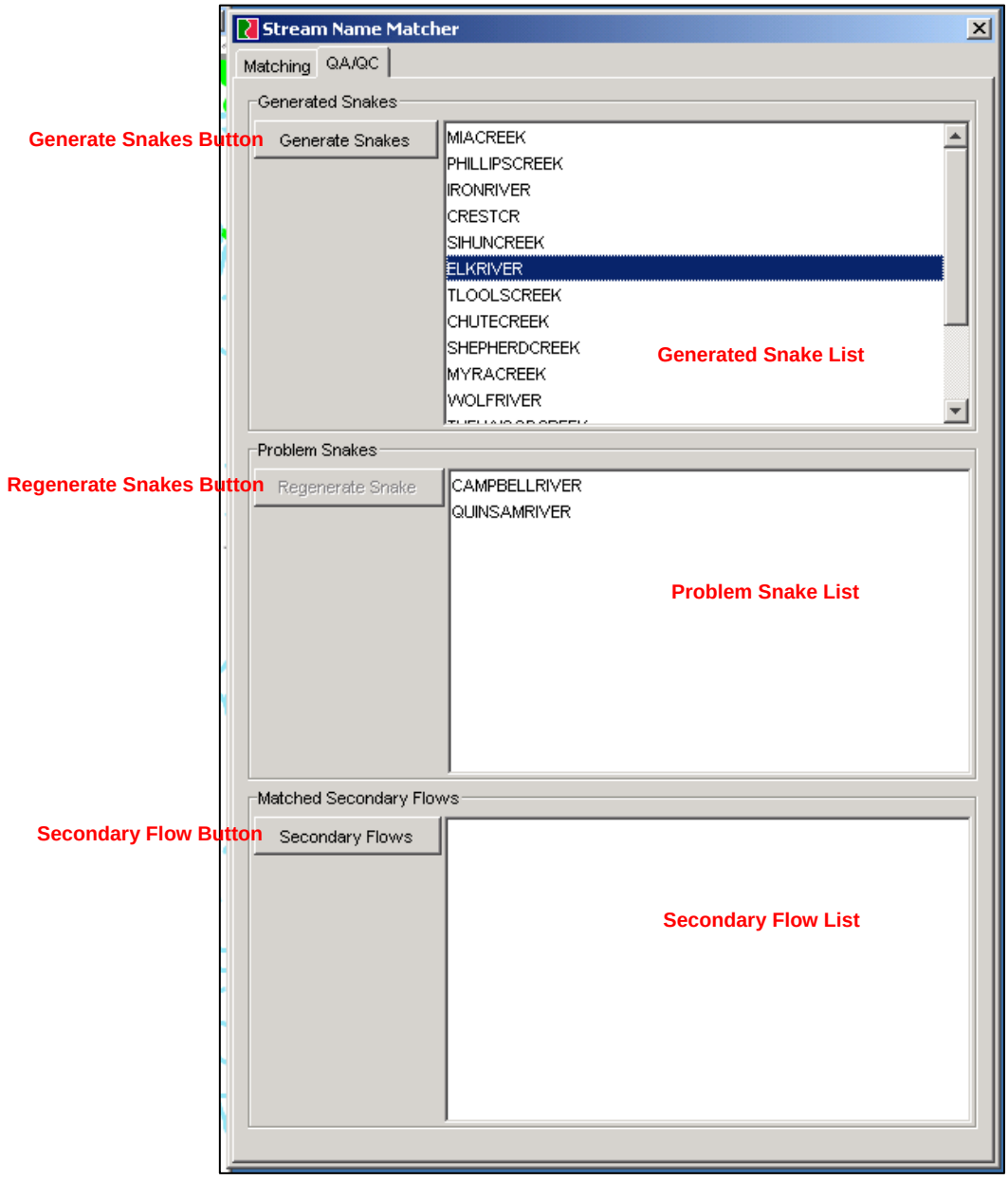
Each set is termed a “worm” as it is small and appears to crawl around the stream system. As seen in the diagram above, there are multiple worms, each of which represent the name “ELK LAKE”. What we would like to do is join these three sets of worms together into a single set as shown below.



In the figure above, the dark blue segments represent the original “ELK LAKE” worms. The red segments represent the edges needed to join together the worms. Together these segments form a continuous set of named segments, which we term a “snake”. The processing of stringing the worms together is known as “snaking”.

5.2 Snake Generation Tool

The Stream Toponym Matcher includes a tool for automatically generating snakes. Its intended use is after all of the name matching has been completed. Shown below is the snake generation panel.



5.2.1 Generate Snakes Button

This button starts the automatic generation of snakes.

5.2.2 Generated Snakes List

This list contains all the snakes that were successful. Selecting an item in the list centers the task view panel around the snake.

5.2.3 Problem Snake List

This list contains all the snakes that could not be formed. Selecting the item in the list centers the task view panel around the worms that could not be formed into a snake.

Two segments are also highlighted. These segments are the reason why the snake could not be formed. It is likely that there is a mistake in the naming. See section 4 for details on possible naming errors.

5.2.4 Regenerate Snake Button

After the user has fixed the problem surrounding the snake generation, this buttons again attempts to regenerate the snake.

5.2.5 Secondary Flow Button

This button starts a process that searches for any secondary flows that have been named. (See section 4)

5.2.6 Secondary Flow List

If any secondary flows that have been named are found, it shows up in this list. Selecting a value from the list centers the task view panel around the named secondary flow.