RoadMatcher

User Guide

RoadMatcher Version 1.4

Prepared by:

Table of Contents

1 Introduction

Conflation of linear road networks involves matching segments from two source networks, each a representation of the same geographic area. The more authoritative segments from the networks are selected for inclusion in the result network and their geometries are adjusted as necessary. The conflation process produces an authoritative, topologically consistent road network.

The RoadMatcher application provides a complete environment for carrying out the conflation of linear road networks. It provides conflation functions such as matching and geometry adjustment in both automated and human-assisted modes. Automated matching and adjustment reduce the amount of manual effort required to conflate large datasets. Rich cartographic visualization facilitates the comprehension of complex conflation scenarios. Statistics are maintained to track the progress of conflation sessions. An issue log is supplied to allow reporting and tracking of anomalous data situations.

Although primarily intended for conflation of road networks, RoadMatcher can be applied to any linear network of man-made features.

This document provides an overview of road network conflation terminology and workflow, of the JUMP Unified Mapping Platform (JUMP) on which RoadMatcher runs, and of the features and functions of RoadMatcher itself. Several road-matching examples are supplied to illustrate the use of the application.

Intended Audience

This document is primarily intended for users of the RoadMatcher application. It will also be useful to any persons who wish to understand the functions and capabilities of RoadMatcher.

Other Documents

RoadMatcher Quick Reference	A summary of RoadMatcher buttons, menu items, and keyboard shortcuts.
RoadMatcher Technical Design	A design for an application that performs road-network conflation.
RoadMatcher Application Requirements	Requirements for the road-network conflation process, intended to describe the process in enough detail to allow a system to be built that supports the automated and human-assisted conflation of road networks in a production environment.
RoadMatcher Conflation Examples	Conflation examples, with emphasis on the resolution of more complex matching situations that are beyond the scope of the AutoMatching algorithms.
JUMP Workbench User's Guide	Instructions on how to use the features of the JUMP Unified Mapping Platform (JUMP) on which RoadMatcher runs

2 Installation

The following system specifications are recommended:

- Pentium III / 1 GHz processor
- 512 Mb RAM

You will need to install two things: Java (if you do not already have it) and the RoadMatcher application.

Installing Java

Before running RoadMatcher, check that you have Java 1.4 or later by opening a Windows Command Prompt and typing "java -version". To download a copy of Java, visit http://java.com.

Installing RoadMatcher

The RoadMatcher application package is distributed as a .zip file that must simply be unzipped and copied to the location of choice on your computer.

(You may have as many instances of RoadMatcher installed on your computer as you wish, provided that each instance is in a separate directory.)

Starting RoadMatcher

1) Locate the *bin* folder in the RoadMatcher installation directory.
2) Open the *bin* subfolder and locate the file *RoadMatcher.bat*.
3) Double-click *RoadMatcher.bat*.

Note: If you open RoadMatcher.bat in a text editor you will see that it is given 256 MB of memory. The minumum recommended memory setting is 128M. The maximum recommended is 80% of the total RAM available.

3 Overview of Road Network Conflation

Road network conflation involves the comparison of two digital road network datasets and the **integration** of the best or most authoritative roads from each into a single higher quality **result network**. To accomplish this, segments in one source network must be **matched** to equivalent segments in the other. For each match, one of the two segments involved will possess geometry and/or attributes that determine it as the preferable or higher quality segment. The preferred segment is marked as the **reference** segment, and will be **included** in the result network. To facilitate one-to-one matching, in some cases **split-nodes** will have to be introduced into segments. In some cases no match will exist for roads, and the roads can be **included** directly as **standalone** segments. In other cases, some roads should not appear in the result and will be **retired** from consideration in the conflation process. The roads included in the result network may require some geometric **adjustment** in order to create **consistent** network topology. When all segments in the source networks have been conflated, the reference roads can be exported to a final, authoritative result dataset.

The goal of a computerized conflation system is to automate as much of the work of conflation as possible. Sophisticated algorithms are used to determine matches and adjustments automatically. In some cases, the two source datasets will be very similar; in others, they will differ greatly. Because of this variation in input quality, roads conflation will almost always involve both automatic and manual (entirely operator-driven) operations.

3.1 Road Network Data Model

It is important to be familiar with the following definitions and concepts.

Road Network

A Road Network is a collection of **segments**, joined to each other at **nodes**.

Source Road Networks

During a road-matching session, you will be working with two existing road network datasets, called **source road networks**.

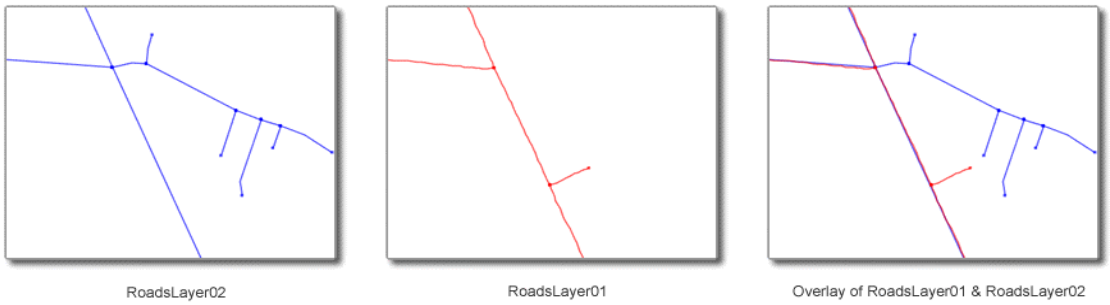


Figure 3-1 – Two source road networks

Result Road Network

The conflation of two source networks produces a new **result network**, which contains the preferred values of the segment geometry and attributes for the area being conflated.

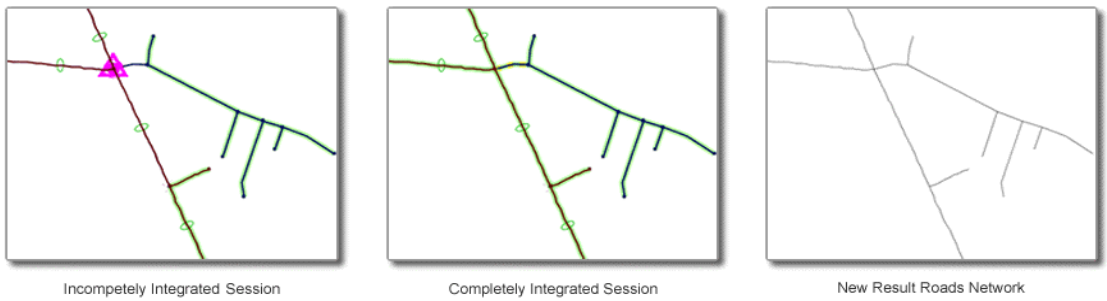


Figure 3-2 – Fixing an inconsistency before exporting the result

Segments

A **segment** is the smallest unit of road represented in a road network. A segment is the length of road between two end **nodes**. Segments are the basic unit that is matched in the RoadMatcher. To aid in matching, a segment may be split into two or more sub-segments via the addition of **split nodes**. After matching, segments may have their geometry **adjusted** in order to create topological consistency in the result network.

Attributes

A segment may have a set of **attributes** attached to it. Attributes are name-value pairs, which contain information about the road such as identifier, name, type, surface type, etc. Attributes can be used to aid in making decisions about how to conflate segments. For matched segments, the attributes will be merged and copied into the result network as the attributes of the result segment.

Nodes

Nodes are the points that define the termini of segments. Start- and end-nodes are the only places at which segments may come into contact with one another.

Vertices

A **vertex** is any point in the definition of the linear geometry for a segment. Typically vertices are used to indicate changes in direction of the geometry of a segment. The first and last vertices of a segment occur at the end nodes of the segment.

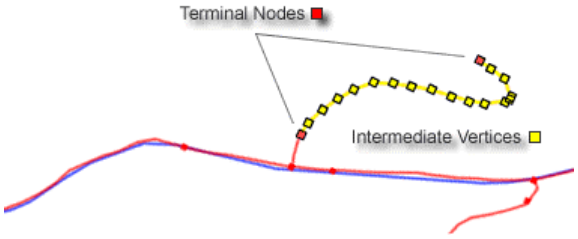


Figure 3-3 – Nodes and vertices

3.2 Segment States

Through the conflation process, RoadMatcher roads progress through a series of states as they are being integrated into the result network.

3.2.1 Source States

Source states are the states that source segments pass through as their level of participation in the result network is determined. The following diagram shows the possible source states of a segment:

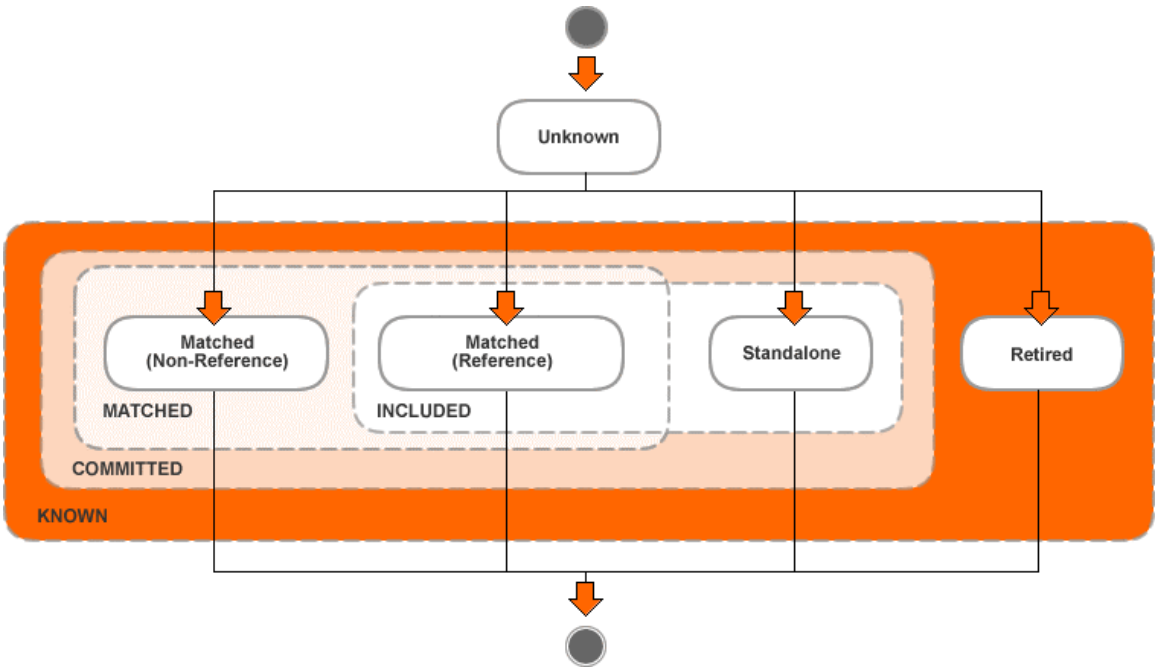


Figure 3-4 – The five source states, their relationships, and the state groups into which they are classified.

There are five source states:

Unknown

The **Unknown** state indicates that there is no information yet about how a source segment should be conflated. At the start of a conflation session all segments are Unknown. After AutoMatching, roads that could not be automatically matched by the application are left in the Unknown state. During the manual matching phase all Unknown roads should be moved to one of the other source states. At the end of the conflation process there should be no roads remaining in the Unknown state.

Standalone

The **Standalone** state indicates that a segment appears in only one of the source networks and is to be included in the result network. Standalone roads are not considered to be matched.

Matched (Reference)
The **Matched (Reference)** state is for a segment that has a matching segment from the other source network and that has the more authoritative or preferable geometry. A Matched Reference segment will be included in the result network.

Matched (Non-Reference)
The **Matched (Non-Reference)** state is for a segment that has a matching segment from the other source network but that has the less preferable geometry. A Matched Non-Reference segment will **not** have its geometry included in the result network – just its attributes.

Retired
The term **Retired** describes a segment, in either source network, that will not be included in the result road network. This can occur when real-world roads are re-routed or decommissioned. It can also occur because a segment was present in a dataset only to define a particular network topology that is no longer relevant.

For convenience, the source states are also grouped into the following **state groups** (see the preceding figure):

Known
The **Known** state group contains the Standalone, Matched (Reference), Matched (Non-Reference) and Retired states. It describes segments that are no longer Unknown.

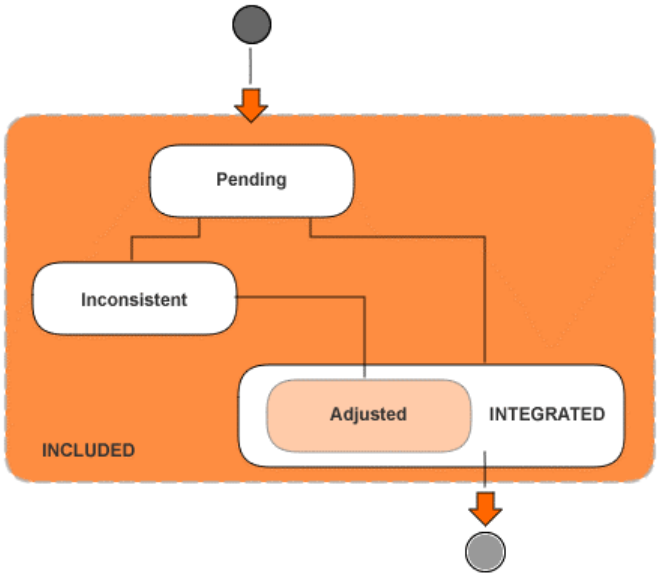
Committed
The **Committed** state group contains the Standalone, Matched (Reference) and Matched (Non-Reference) states. It describes segments that will contribute geometry or attributes to the result network.

Matched
The **Matched** state group contains the Matched (Reference) and Matched (Non-Reference) states. It describes segments that participate in a match, as either the reference or the non-reference.

Included
The **Included** state group contains the Standalone and Matched (Reference) states. It describes segments whose geometries will be included in the result network (possibly with subsequent geometry adjustment).

3.2.2 Result States

Segments that have been Included in the result network are further categorized into a set of **Result States**. The following diagram shows the result states and how they are related:



The three result states are:

Pending

The **Pending** state indicates an Included (Matched or Standalone) segment whose topological consistency in the result road network cannot yet be determined. A segment's consistency can only be determined when all the segments adjacent to it are in a Known state. Once all neighbours of a segment are marked as Known, RoadMatcher will automatically move the segment into the Inconsistent or Integrated state.

Inconsistent

The **Inconsistent** state indicates a segment whose network topology is for some reason inconsistent. Inconsistency can occur due to:

- an intersection between segments not at nodes
- a discrepancy between the topology of matched segments.

Inconsistencies can be fixed by adjusting result geometry using the node and vertex adjustment tools described in *6 Manual Conflation*.

Integrated

The **Integrated** state indicates an Included segment that has been consistently incorporated into the result road network. The goal of the conflation process is to move all segments in the result network into the Integrated state.

The **Adjusted** sub-state indicates a segment whose geometry has been altered to facilitate its integration into the result network. Generally this will be done in order to resolve inconsistencies.

When segments are marked as Included (Matched (Reference) or Standalone) RoadMatcher will automatically determine whether they can be Integrated. The topology and geometry of some segments will allow them to be integrated into the result network immediately. Other segments will have inconsistent topology and will need to be adjusted before they can be Integrated (see *6.5 Adjust Endpoint Tool*).

3.3 Match Distance Metrics

There are several distance metrics which are useful for determining how well two segments match. RoadMatcher computes these metrics automatically when segments are matched, and allows them to be viewed for QA purposes.

3.3.1 Maximum Distance

The **Maximum Distance** between two segments is a measure of how far apart the segments are. It corresponds to the maximum of the distances from all points on either segment to the other segment. (Technically speaking, this is the **Hausdorff distance** between the segments.)

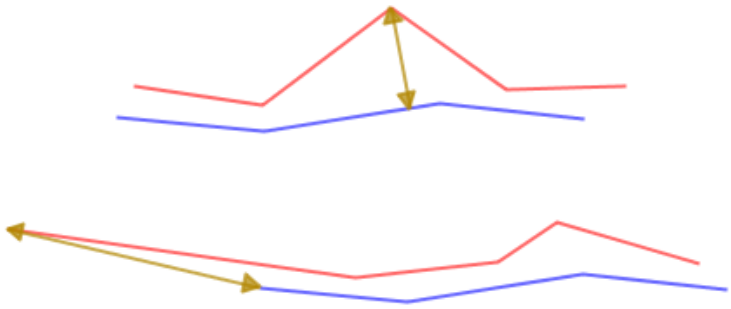


Figure 3-5 – Maximum Distance between matched segments

3.3.2 Trimmed Distance

The **Trimmed Distance** is designed to give a better distance metric for matched segments which are very different in length. It uses the same distance metric as the Maximum Distance, but before computing it trims the segments to remove any unmatched length from the segment ends.

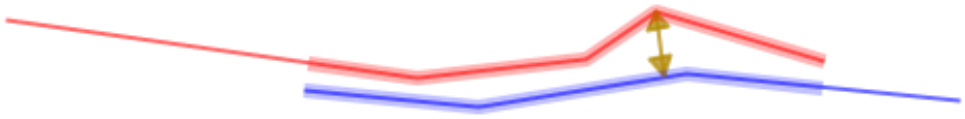


Figure 3-6 – Trimmed Distance between matched segments

3.3.3 Nearness

The **Nearness** of matched segments is a fraction indicating how much length of the segments is within a given **Nearness Tolerance** of the other segment. The Nearness value is useful for QA purposes because unlike the other distance metrics it takes into account the entire length of the segments. The Nearness is computed on the trimmed segments. The Nearness Tolerance may be determined from pre-existing data quality parameters, or can be set from statistics gather empirically (e.g. the average Trimmed Distance between matches in similar datasets.)

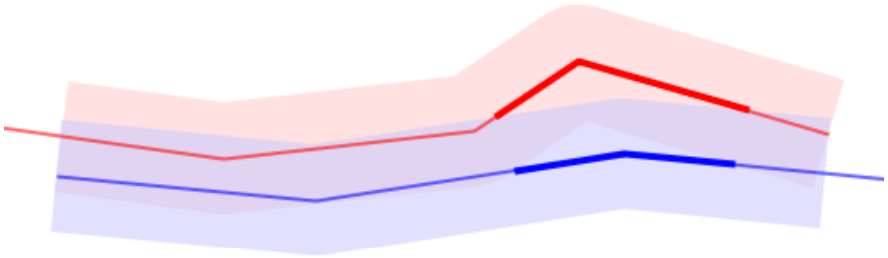


Figure 3-7 – Nearness of matched segments

3.4 Distributed Conflation

Sometimes it is necessary to perform conflation of a dataset in a distributed fashion. This means that the conflation area is split into two or more regions, and the data in each region is conflated independently. This can be required for reasons of performance, dataset size, or workflow scheduling. Often it may be different operators performing the conflation of each data subset, at different times, with limited or no communication between them. It is important that the conflated data be consistent across region boundaries, to allow the regions to be merged correctly into a single conflated dataset.

RoadMatcher provides two features to help ensure inter-region consistency during distributed conflation: **Node Constraints** and **Context Layers.**

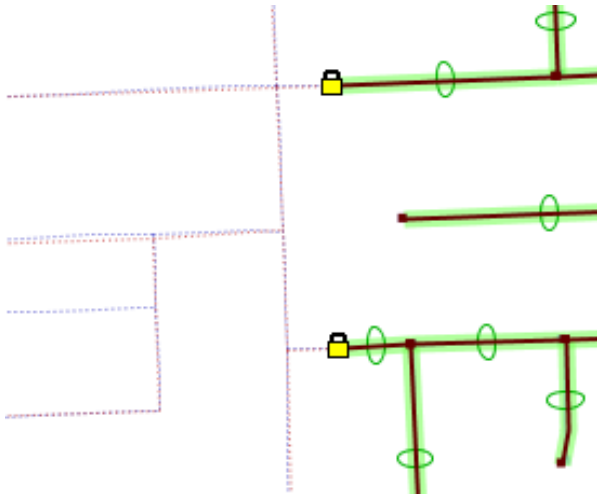


Figure 3-8 Example showing Node Constraints and Context Layer data

Node constraint data and context layer data are most easily supplied to sessions created from input packages. It can also be specified when sessions are created using the New Session wizard.

3.4.1 Node Constraints

Node Constraints allow nodes to be "locked in place". The software will ensure that they cannot be moved during the conflation process. Constraining nodes ensures that segments which cross region boundaries match segments in the neighbouring region exactly.

Node Constraints are displayed with a "lock" symbol as in the illustration.

Note: it may be necessary to insert nodes into the source datasets where segments cross dataset boundaries. The nodes must be inserted by pre-processing the datasets. After conflation, these nodes can be removed by post-processing.

3.4.2 Context Layers

Context layers provide background linework to assist making appropriate matching and adjustment decisions during the manual conflation process. The data on these layers cannot be selected or changed. However, the geometry on the layers can be used to snap to during adjustment.

Context layers are particularly useful during distributed conflation, to allow operators to see the topology of the source networks in adjacent regions. Of course, context layers may also be useful during conflation of a single area.

3.5 Conflation Attributes

To each segment, RoadMatcher adds attributes containing information about how the segment was conflated. Each of these **conflation attributes** is described below.

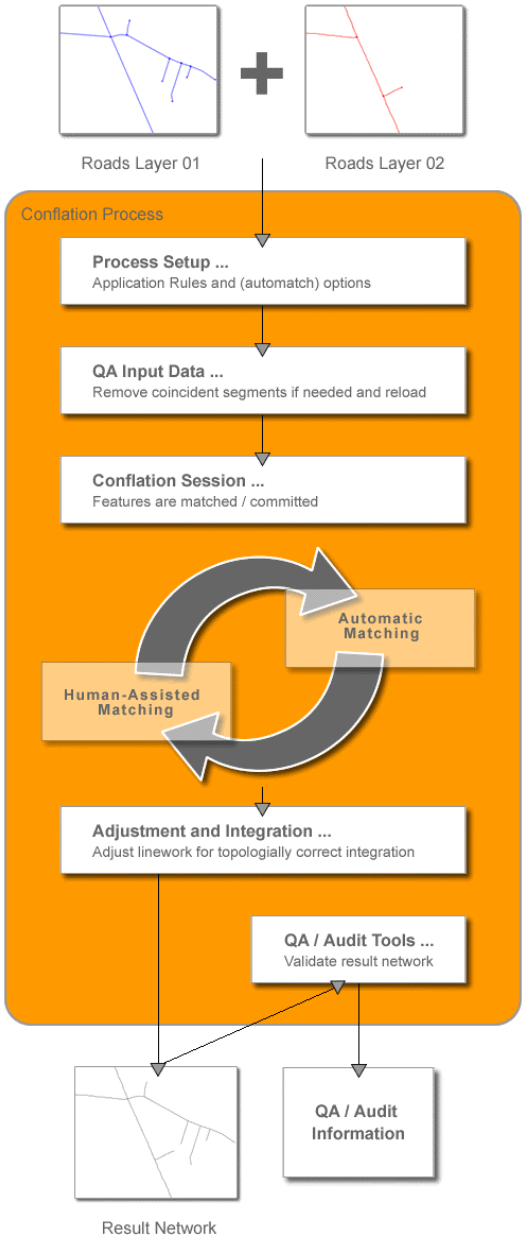
Table 3-1 – The conflation attributes

Max Distance	A measure of how far a segment is from its matching segment. (See *3.3.1* Maximum Distance).
Adjustment Size	The maximum distance between the original and adjusted geometries of a segment. 0 if the segment's geometry is unmodified.
Trimmed Distance	A better distance metric for matched segments which are very different in length. (See *3.3.2* Trimmed Distance).
Length	The length of the current geometry of the segment.
Nearness	A fraction indicating how much length of the segments is within a given Nearness Tolerance of the other segment. (See *3.3.3* Nearness).
Comment	A user-entered comment for this segment. (See *10.4* Segment Comments).
Adjusted	A Boolean indicating whether the geometry has been adjusted.
Adjustment Angle Delta	The change in angle of the adjusted segment, in degrees. "Angle" here refers to the angle of a straight line from the start node to the end node of the segment. 0 if the segment is not adjusted.
Source State	One of: **Unknown**, **Matched (Reference)**, **Matched (Non-Reference)**, **Standalone** or **Retired** (see 3.2.1 Source States).
Result State	One of: **Pending**, **Inconsistent**, or **Integrated** (see 3.2.2 Result States).
Split Start	A Boolean indicating whether the start node is a split node (see 6.4 Split Node Tools).
Split End	A Boolean indicating whether the end node is a split node.
Manually Matched	A Boolean indicating whether the match was made using the manual conflation tools rather than AutoMatch
Manually Adjusted	A Boolean indicating whether the adjustment was made using the manual conflation tools rather than AutoAdjust
Manual Adjustment Count	The number of adjustments made using the manual conflation tools
Match Orientation	The relative orientation of matched segment geometries. If true, the geometries match in their direction of digitization. If false, the geometries are oriented in different directions in the match.
Reviewed	A Boolean indicating whether the segment has been marked as Reviewed (see *10.5* Marking Segments As Reviewed).
Update Time	The last time the segment's geometry or state changed.

The conflation attributes described above may be examined using the **Query** window (see Section 10.2).

4 Overview of Conflation Workflow

The diagram on the right summarizes the workflow of the conflation process.



Before segments are matched, several processing parameters are set. These include the short names, matching precedence, and adjustment constraints – see *5.2* Creating a Session from a Source Package and *5.3* Creating a Session.

At the beginning of a conflation session, if RoadMatcher detects coincident segments or illegal geometries, these errors must be corrected before you begin matching. These errors are discussed in *5.1.1* Coincident Segments and *5.1.2* Illegal Geometries.

With two valid source datasets loaded and all of the RoadMatcher parameters set, you are ready to begin your conflation session. Conflation sessions will usually start with AutoConflation, wherein the tools of the RoadMatcher application automatically conflate as many segments as the parameters allow (see *7* Automated Conflation). Automated conflation will usually be followed by human intervention in which any Unknown segments are marked as Matched or Standalone, and Inconsistent segments are adjusted to correct the inconsistent topology (see *3.2.1* Source States).

Note: This user guide will discuss manual conflation before automated conflation, in order to introduce you to the basic RoadMatcher operations first.

After an AutoMatch, you must resolve all Unknown and Inconsistent segments. Any remaining Unknown and Inconsistent segments can be located using RoadMatcher (see *10.2* Query), and must be dealt with so that no segments are Unknown or Inconsistent before the result dataset is exported (see *9* Exporting the Result).

5 Starting a Conflation Session

This section describes how to get started, beginning with requirements for the input data (Section 5.1). Next are discussed the two methods for starting a conflation session: from a source package (Section 5.2) and from scratch (Section 5.3).

5.1 Data Requirements

The RoadMatcher application expects to operate on source datasets with linear network topology. Linear network topology implies that the geometry of the dataset fulfills the following conditions:

- All segments have linestring geometry (no multilinestrings allowed)
- Segments are considered to connect only if they terminate at a shared node (i.e. their endpoints share mathematically equal endpoints; no linestring-crossing allowed)
- Isolated segments are allowed

5.1.1 Coincident Segments

The term **coincident segments** refers to a situation in which two segments that are incident on the same node have coincident geometry at that node. In other words, the linework for the segments is parallel (has the same angle) where it meets the node. This situation prevents RoadMatcher from determining correct network topology. It is flagged as an error, and should be corrected before continuing with the conflation process. To fix the error, you can use an external editor like ArcView or JUMP's built-in editing tools (see the *JUMP User Guide*).

The following diagram illustrates the concept of segment coincidence:

Figure 5-1 – Road with two coincident line segments on the left (circled)

If RoadMatcher detects coincident segments in the source datasets, it will handle them as described in *5.1.4 QA of Source Datasets*.

5.1.2 Illegal Geometries

Besides coincident segments, another serious error in the input datasets is that of **illegal geometries**. Each segment must be spatially represented with a **linestring** – other geometry types are not allowed e.g. points, polygons, multilinestrings, etc. Moreover, linestrings must **not self-intersect**. See Figure 5-2 below.

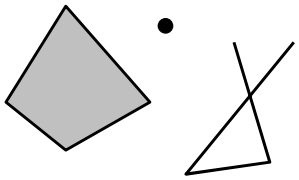


Figure 5-2 – Examples of illegal geometries for segments

If RoadMatcher detects illegal geometries in the source datasets, it will handle them as described in *5.1.4 QA of Source Datasets*.

5.1.3 Other Issues

The following situations will not cause the source datasets to be marked as invalid, but should be avoided so that the result dataset has good network topology.

Table 5-1 – Miscellaneous topology problems

Issue	How to find	How to fix
Intersecting segments 	Zoom out, look for purple triangle markers	**Create Intersection SplitNode** menu item (see *6.4* Split Node Tools)
Short segments. Note that we are concerned here with vertex-vertex distance, not the length of the entire segment. 	**Tools > QA > Validate Selected Layers > Check minimum segment length**	Vertex Editing Tools (Section 6.7)
Very close nodes	(no built-in function)	**Adjust Endpoint Tool** (Section 6.5)

5.1.4 QA of Source Datasets

When a conflation session is started RoadMatcher checks for the presence of coincident segments and illegal topologies (see *5.1.1 Coincident Segments* and *5.1.2 Illegal Geometries*). If any are found in the input datasets, a message like the following is displayed:

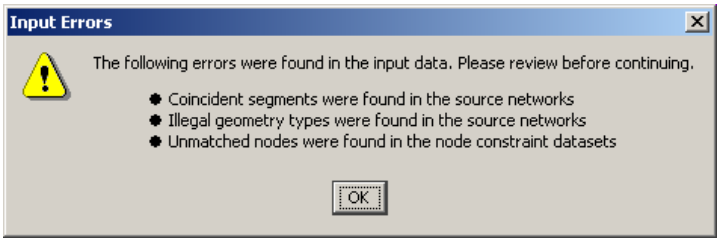


Figure 5-3 – Input Errors

Coincident segments found in the source datasets are placed in a **Coincident Segments** layer in the **QA** folder (to the left of the map); illegal geometries, in an **Illegal Geometry** layer. The **QA** folder appears only if coincident segments or illegal geometries were found. To fix the error, you can use an external editor like ArcView or JUMP's built-in editing tools (see the *JUMP User Guide*).

The third error shown above has to do with node constraints. Node constraints without matching nodes are placed in an **Unmatched Node Constraints** layer. For more information on node constraints, see Figure 5-8 – Selecting node-constraint layers.

Note: RoadMatcher will let you carry on even if coincident segments or illegal geometries are present in the source datasets. But if you leave coincident segments unfixed, you may encounter errors when trying to match or adjust them. The following note will appear in the upper-right corner as a reminder:

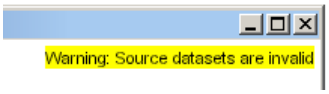


Figure 5-4 – "Source datasets are invalid" warning in the upper-right corner

5.2 Creating a Session from a Source Package

An easy way to get data into RoadMatcher is to import a **source package**, if you have one available. (If you do not have a source package, follow the steps outlined in *5.3* Creating a Session from Scratch).

To create a session from a source package, simply click **RoadMatcher > New Session From Source Package** and specify the source-package filename.

Next, you will be asked whether you want to run an AutoMatch and AutoAdjust on the data:

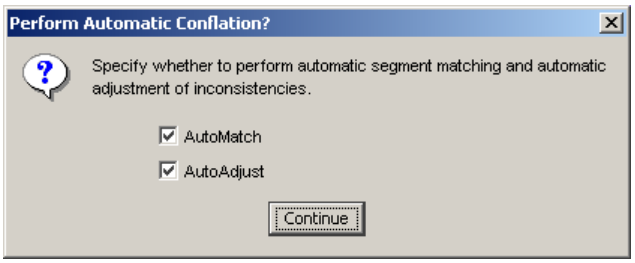


Figure 5-5 – Specifying whether to run an AutoMatch and AutoAdjust

Typically you will want to run an AutoMatch and AutoAdjust, to match and fix most of the data automatically. The exception is if your source package does not contain a profile (*.rmprofile) – in this case you may want to wait until you've set the parameters for AutoMatch and AutoAdjust before running them (see *11.2 AutoMatch Options* and *11.4 AutoAdjust Options*).

Note: The boxes will automatically be checked or unchecked depending on whether or not your source package contains a profile.

A source package contains everything you need to begin your conflation session: the two datasets, their short names, match precedence rules, AutoMatch options, adjustment constraints, AutoAdjust options, and result options (see *12* Creating a Source Package).

5.3 Creating a Session from Scratch

This section describes how to create a RoadMatcher session for two datasets. (If your datasets are contained in a **source package**, you can save a few steps by skipping this section and going to *5.2* Creating a Session from a Source Package).

Before creating the conflation session, ensure that your two datasets are loaded. Use **File > Load Dataset(s)** to load them from Shapefiles, GML files or Well-Known Text. For more information on loading datasets, see the *JUMP User Guide*.

Next, click **RoadMatcher > New Session**. You will see the wizard below:

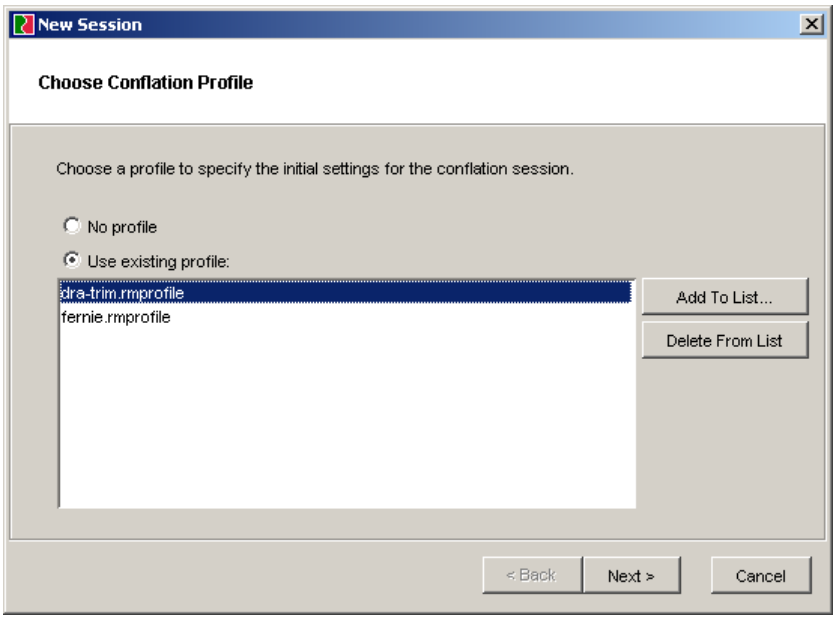


Figure 5-6 – Choosing a conflation profile

Here you choose a **profile**, which is a file containing predefined settings for various conflation parameters: short names, match precedence rules, AutoMatch options, adjustment constraints, AutoAdjust options, and result options. If you do not have any profiles, choose **No profile** to accept the default values. (You can change these values later – see *11.1* Match Precedence*, 11.2* AutoMatch Options*, 11.3* Adjustment Constraints, *11.4* AutoAdjust Options*, 11.5* Result Options).

Note: For information on creating a profile that you or others can use in future conflation sessions, see *11* Defining a Conflation Profile.

Then press **Next**. You will be prompted to specify the two input datasets:

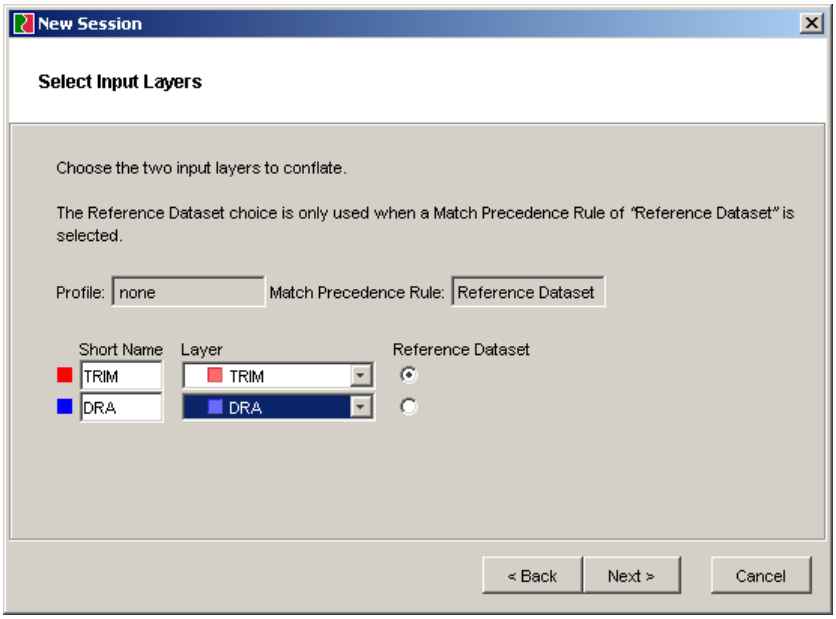


Figure 5-7 – Selecting the two input layers

Click the **Layer** drop-downs to pick the two datasets you loaded earlier. If you selected **No profile** on the previous screen, you can enter a **short name** for each dataset. These short names will be used to refer to the datasets throughout RoadMatcher.

Depending on which profile you chose on the previous screen, you may or may not see a **Reference Dataset** indicator. First, some background – When conflating two segments, one segment's geometry must be chosen over the other. The chosen segment is termed the **reference**, while the other is the **non-reference**. Both segments will contribute their attributes to the final, **result** segment, but the geometry will come from the reference segment only. (For more information, see *3.2.1* Source States).

During AutoMatching, RoadMatcher will choose the segment from the Reference Dataset as the reference. (However, during manual matching, you are free to choose which of the segments will be the reference).

If you do not see a **Reference Dataset** indicator, you have chosen a profile that specifies a custom rule for choosing the reference during AutoMatching – custom rules will be discussed in *11.1* Match Precedence.

Press **Next** – you will be prompted to specify **node constraints**, if any:

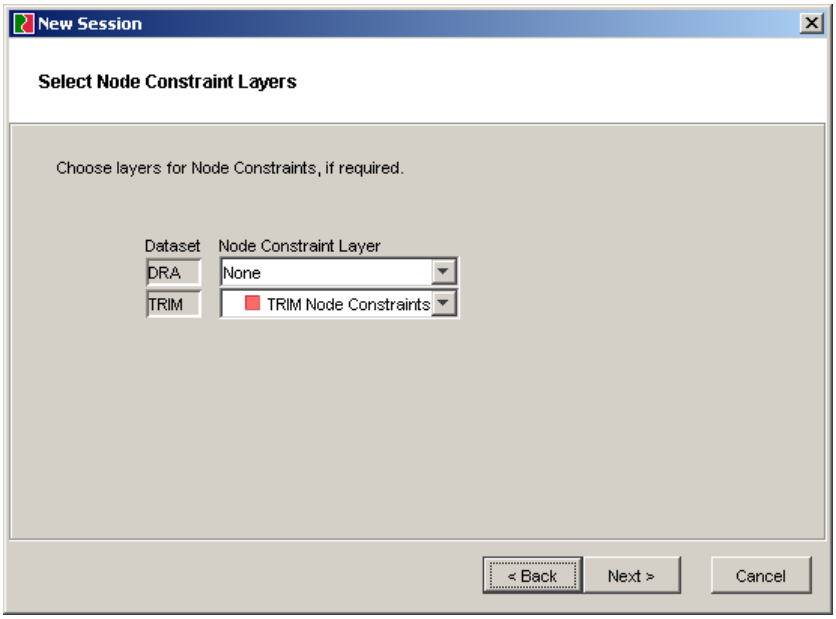


Figure 5-8 – Selecting node-constraint layers

A node-constraint layer contains points specifying which endpoints cannot be changed during the conflation process. Node constraints ensure that networks which are subsets of larger network databases can be conflated and still maintain connectivity to the larger network.

Specifying a node-constraint layer is optional. If you do not have any node-constraint layers, leave them set to "None".

Press **Next** – you will be asked whether you would like to run an **AutoMatch** (Section 7.1) and an **AutoAdjust** (Section 7.2):

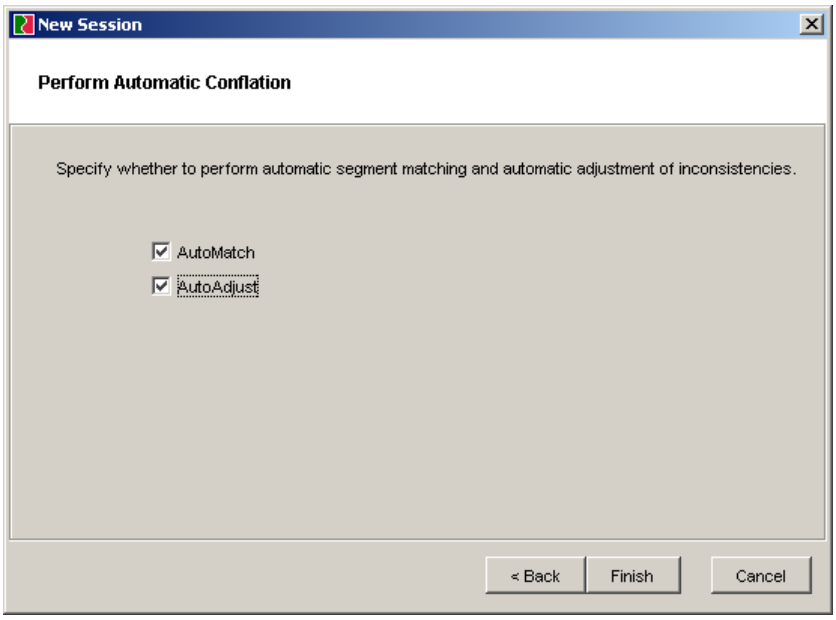


Figure 5-9 – Specifying whether to run an AutoMatch and AutoAdjust

Typically you will want to run an AutoMatch and AutoAdjust, to match and fix most of the data automatically. The exception is if you did not choose a profile (on the first screen) – in this case you may want to wait until you've set the parameters for AutoMatch and AutoAdjust before running them (see *11.2 AutoMatch Options* and *11.4 AutoAdjust Options*).

Note: The boxes will automatically be checked or unchecked depending on whether or not you chose a profile on the first screen.

Finally, press **Finish** – RoadMatcher will start a new conflation session with the settings you specified.

6 Manual Conflation

Typically you would run automated conflation first before using the manual conflation tools. However, we will discuss manual conflation first to give you an understanding of basic conflation techniques. Automated conflation will be covered in Section 7.

Note: Whenever you use one of the manual matching tools, RoadMatcher will AutoAdjust (Section 7.2) the segments to automatically fix simple inconsistencies. To turn this feature off, see *10.10* Application Options.

Note: You may at times be warned about segment having "a line segment shorter than the minimum defined in the AutoMatch options" or "line-segment length < min length". This warning indicates that one of the segment's straight-line segments is shorter than the minimum tolerance. To learn more about this tolerance, see *11.2 AutoMatch Options*.

6.1 Match/Commit Tools

With the **Match/Commit Tools**, you can establish a match between a pair of segments; you can also mark a segment as Standalone. Semi-automated matching tools will be described in later sections (*6.2* Path Match Tool and *6.3* Define Paths Tool).

The main **Match/Commit Tool** is circled in the figure below:

Figure 6-1 – The Match/Commit Tool (circled)

The basic usage of this tool is to drag the cursor across two segments to match:

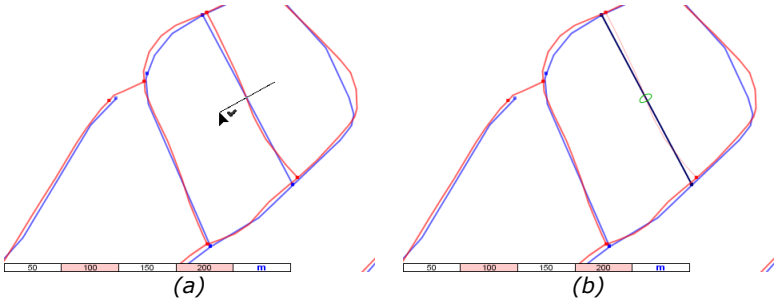


Figure 6-2 – (a) During and (b) after the use of the Match/Commit Tool

The green ring in Figure 6-2b indicates that the two segments dragged across are now matched. (Any Retired segments that you drag across will be ignored).

Which of the two segments did RoadMatcher choose as the reference, and which as the non-reference? It depends on the match precedence rules – for more

information, see *11.1* Match Precedence Rules. You can override match precedence rules by using one of the dataset-specific **Match/Commit Tools**:

Figure 6-3 – The dataset-specific Match/Commit Tools (circled)

The red tool on the left will establish the match, but will mark the **red** dataset as the reference and the blue dataset as the non-reference. In contrast, the blue tool on the right will establish the match, but will mark the **blue** dataset as the reference and the red dataset as the non-reference. For more information on the Matched (Reference) and Matched (Non-Reference) states, see *3.2.1* Source States.

Sometimes it will be difficult to make a match by dragging the cursor across two segments because another segment lies between them. In this case, instead of dragging the cursor, simply **click** the first segment, then click the second. Alternatively, use the **Select Features Tool** to select the two segments, then click **RoadMatcher > Match Selected Segments**.

The **Match/Commit Tools** are also used to mark isolated segments as Standalone, as shown below.

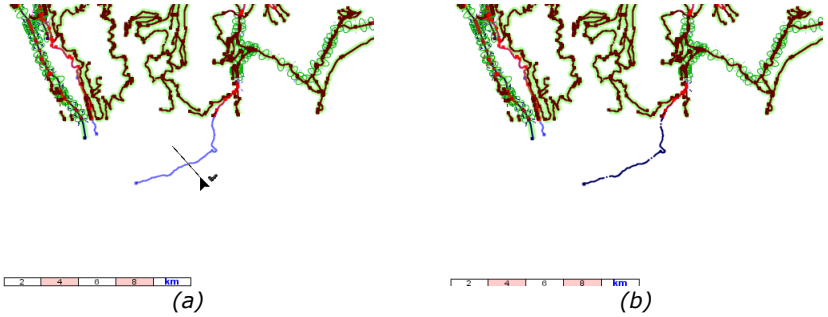


Figure 6-4 – (a) During and (b) after the use of the Match/Commit Tool to mark a segment as Standalone

In Figure 6-4, note that the segment has assumed a darker shade of blue, indicating that it is now marked as Standalone (see *10.3* Legend).

Note: Establishing one-to-one matches between segments may require you to split segments – see *6.4* Split Node Tools for more information. Alternatively, try one of the more powerful matching tools (*6.2* Path Match Tool*, 6.3* Define Paths Tool*, 7.1* AutoMatching), which split segments automatically.

Note: To fix simple inconsistencies, an AutoAdjustment (see *7.2* AutoAdjustment) is performed whenever you use the **Match/Commit Tools**. To turn this feature off, see *10.10* Application Options.

6.2 Path Match Tool

The **Path Match Tool** adds an element of automation to manual matching operations. Instead of matching only two segments as we did with the **Match/Commit Tool** (Section 6.1), with the **Path Match Tool** you specify two segments to begin matching on – it will then attempt to continue matching in a linear path, starting at either end.

The **Path Match Tool** is circled below:

Figure 6-5 – The Match/Commit Tool (circled)

Figure 6-6 below demonstrates using the **Path Match Tool** to specify two segments at which to begin matching.

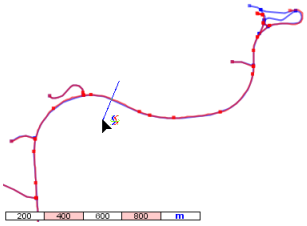


Figure 6-6 – Specifying the initial pair of segments with the Path Match Tool

When you release the mouse button, the **Path Match Tool** matches as many segments as possible, in a linear path:

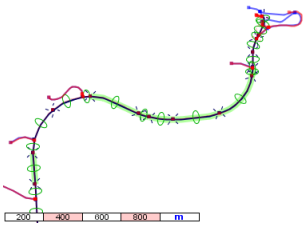


Figure 6-7 – Matches created by the Path Match Tool

While results may vary, note that you are able, with the **Path Match Tool**, to match a large number of segment pairs quickly and with minimal interaction.

Also note that some of the matched segments are marked in green. As described in the Legend (Section 10.3), this means that they are in the Integrated state i.e. they have no problems and are ready to be integrated into the result dataset. For a description of Integrated and other result states, see *3.2 Segment States*.

Note: In some cases, the **Path Match Tool** will fail to find paths that match. Typically this happens because the two paths separate by a distance greater than the **Maximum Match Distance** (see *11.2* AutoMatch Options). You may instead want to try the **Define Paths Tool** (Section 6.3), which is not subject to this restriction; alternatively (but requiring more effort), you can match individual pairs of segments using the **Match/Commit Tools** (Section 6.1).

Note: To fix simple inconsistencies, an AutoAdjustment (see *7.2* AutoAdjustment) is performed whenever you use the **Path Match Tool**. To turn this feature off, see *10.10* Application Options.

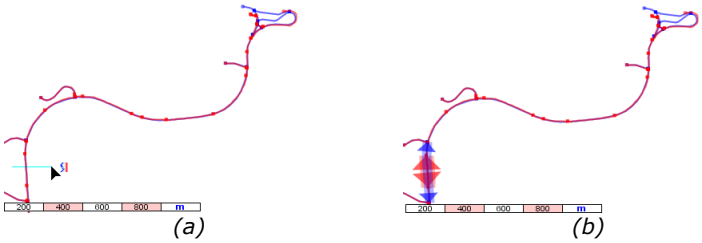
6.3 Define Paths Tool

Like the **Path Match Tool** (Section 6.2), the **Match Paths** function (described in this section) matches segments along two paths, creating split nodes where necessary. However, whereas the **Path Match Tool** automatically finds matching paths – an operation that sometimes fails – the **Match Paths** function allows you to explicitly define the paths using the **Define Paths Tool**, shown below:

Figure 6-8 – The Define Paths Tool (circled)

To begin defining a path, activate the **Define Paths Tool** and drag the cursor across a segment.

Note: You can start defining both paths at the same time by dragging the cursor across two segments – one from each dataset.



(a) *(b)*

Figure 6-9 – (a) During and (b) after using the Define Paths tool to start two paths (one for each dataset) at the lower-left corner

Next, extend the paths by dragging the cursor across two segments that are further along. The **Define Paths Tool** will extend the paths accordingly, as shown below:

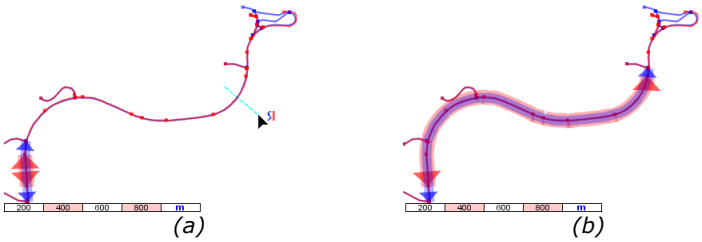


Figure 6-10 – (a) During and (b) after using the Define Paths tool to extend the path

Note: If the path is not obvious to the **Define Paths Tool**, it will be unable to find the segments you specify. In this case, try specifying segments that are closer to the existing path. Or click **RoadMatcher > Clear Paths** to clear the existing paths and start again.

Now that we have defined two paths, we can match them (splitting segments where necessary) by clicking **RoadMatcher > Match Paths**:

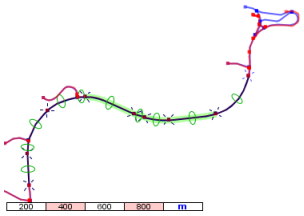


Figure 6-11 – Using the Match Paths function to establish matches along two paths.

If while defining the paths you wish to clear them and start again, click **RoadMatcher > Clear Paths** (or if the **Define Paths Tool** is active, simply double-click anywhere on the map).

Note: If you try to define a path but keep getting the error "Path route cannot be determined", it is likely because a path for that dataset already exists but is off-screen, and the tool is trying to find a route from that path to the segment you are specifying. In this case, simply clear the path (as described above) and start again.

Note: If the Match Paths operation fails (which sometimes happens), you can always resort to using the **Match/Commit Tools** to match individual pairs of segments.

Note: To fix simple inconsistencies, an AutoAdjustment (see *7.2* AutoAdjustment) is performed whenever you use the **Match Paths** menu item. To turn this feature off, see *10.10* Application Options.

6.4 Split Node Tools

When using the **Match/Commit Tools** (Section 6.1), you may find that you need to match a single segment from one dataset with several segments from the other dataset. To achieve one-to-one matches, you will need to split the large segment into several smaller ones. This section describes the splitting of segments i.e. creating **split nodes**.

Note: The **Path Match Tool** (Section 6.2), Match Paths operation (Section 6.3) and AutoMatch operation (Section 7.1) will do the splitting for you automatically.

The six **Split Node Tools** are circled in the figure below:

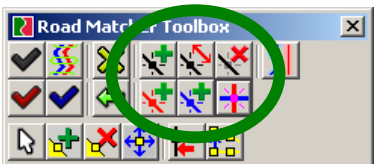


Figure 6-12 – The Split Node Tools (circled)

In the example below, we want to match five red segments to one long blue one. First, we use the **Create Split Node Tool** to split the blue segment into five pieces. Simply activate the tool and click on the blue segment near each red node. At each point you click, a split node will be created.

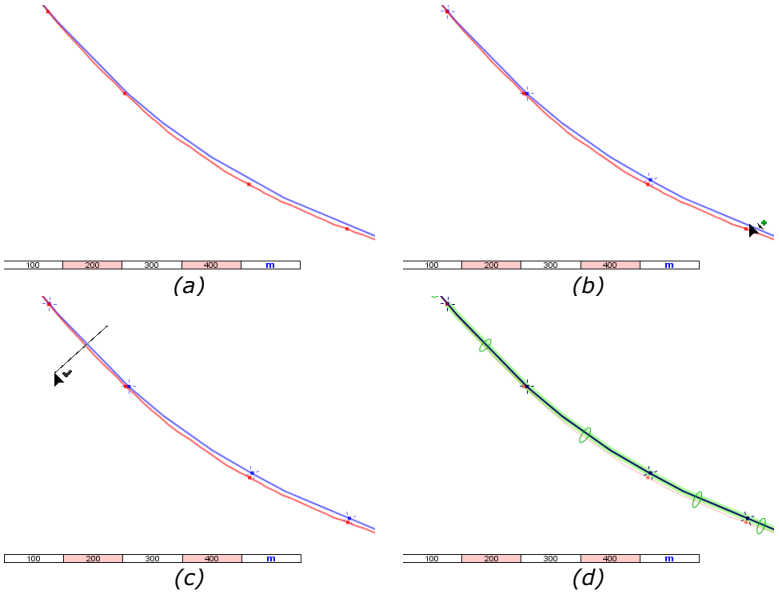


Figure 6-13 – (a) One blue segment is to be matched to five red segments (b) Splitting the blue segment with the Create Split Node Tool (c) Matching one of the segment pairs with the Match/Commit Tool (d) All five pairs of segments are matched.

As shown in Figure 6-13, the main **Split Node Tool** () automatically determines which dataset to insert the split node into (the one that doesn't already have a node

in the vicinity). To explicitly specify the dataset for which to create the split node, use one of the **dataset-specific Split Node Tools** ( , ).

The **Delete Split Node Tool** () is used to remove a split node. Simply activate this tool, then click on the split node to delete. Note that this tool removes split nodes only, not regular nodes (split nodes are marked with crosshairs).

The problem of intersecting segments was mentioned in *5.1.3* Other Issues. This can be remedied using the **Create Intersection SplitNode Tool**. Activate this tool () then click on the intersection – the two segments will be split at the intersection point, resolving the problem:

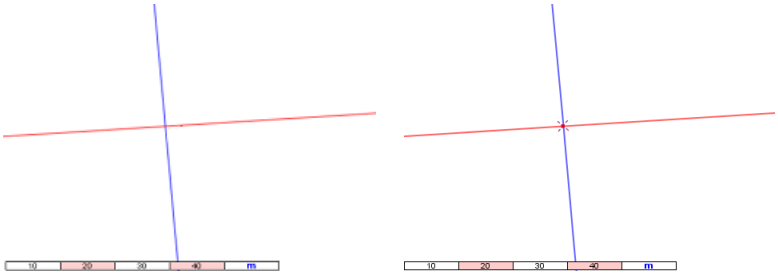


Figure 6-14 – (a) Before applying the Create Intersection SplitNode Tool (b) After. Split nodes have been created on both segments.

Note: The two segments may be from the same dataset or different datasets.

You can also move split nodes. Activate the **Move Split Node Tool** (), then position the cursor over a split node and drag it to a new location along the segment:

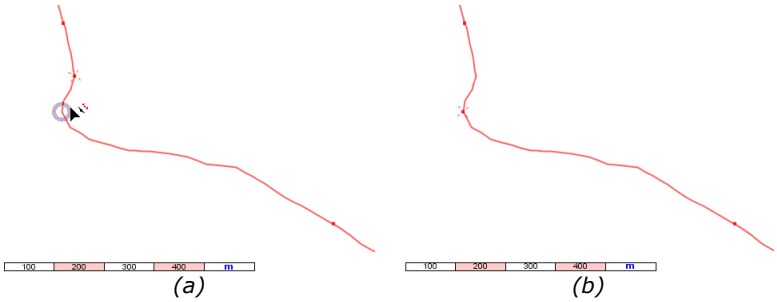


Figure 6-15 – (a) A circle appears as you drag the cursor along the segment (b) When you release the mouse button, the split node appears at the new location.

Note: The **Move Split Node Tool** cannot move regular nodes – just split nodes. For information on moving regular nodes, see *6.5* Adjust Endpoint Tool.

6.5 Adjust Endpoint Tool

With the **Adjust Endpoint Tool**, you can modify the geometry of a segment by dragging one of its endpoints.

Note: A quicker way to resolve simple inconsistencies (albeit with less control) is to use the Auto-Connect Endpoint Tool (Section 6.6).

First select the **Adjust Endpoint Tool**, marked below:

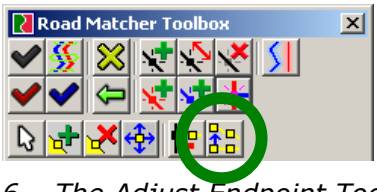


Figure 6-16 – The Adjust Endpoint Tool (circled)

Next, click the **Adjust** tab (on the **RoadMatcher Toolbox**). You will be presented with a choice of five different **adjustment methods**:

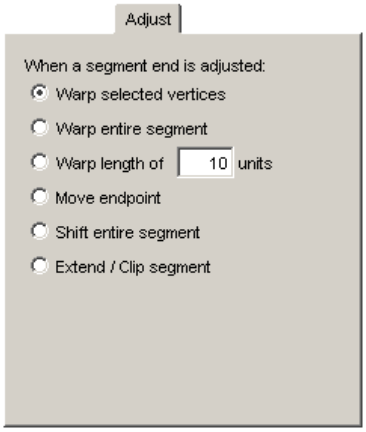
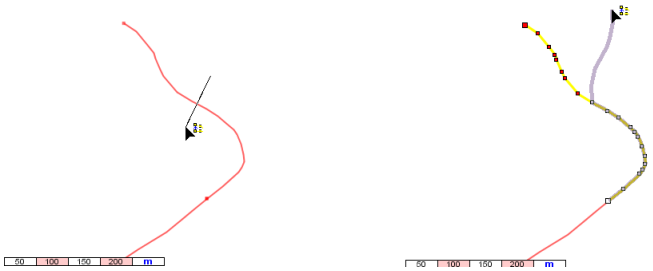


Figure 6-17 – The Adjust tab

The default adjustment method is **Warp selected vertices**. This mode will warp a portion of the segment. To use this adjustment method, first drag the cursor across the segment, drawing a line separating the vertices you want to warp from the vertices you want to leave unchanged. Then drag one of the segment's endpoints to adjust the geometry. This is illustrated below:



(a) *(b)*

Figure 6-18 – (a) Separating the vertices to warp from the vertices to leave alone (b) Dragging the endpoint. Note that only the vertices we specified are involved.

Whereas **Warp selected vertices** is a two-drag operation, the remaining adjustment methods are one-drag operations. They are described next.

Warp entire segment will rotate and stretch the entire segment as you drag the endpoint around:

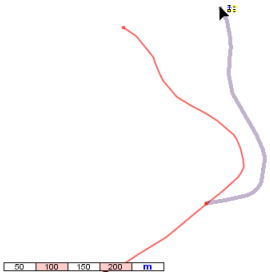
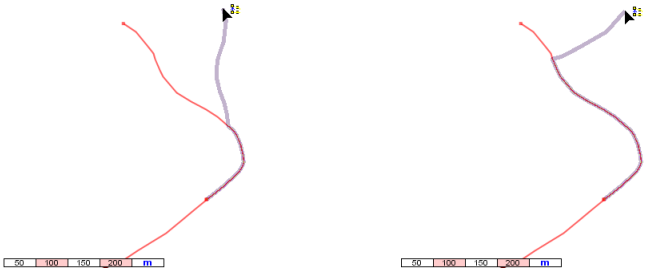


Figure 6-19 – The Warp Entire Segment adjustment method

With **Warp length of X units**, you type in the approximate length of the portion of the segment to warp:



(a) *(b)*

Figure 6-20 – Warping a length of (a) 150 units (b) 50 units

Move endpoint moves just the first vertex (or the last vertex):

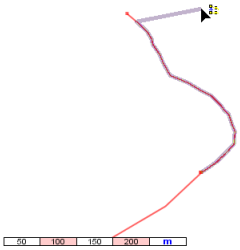


Figure 6-21 – The Move Endpoint adjustment method

Shift entire segment moves the whole segment without changing its shape:

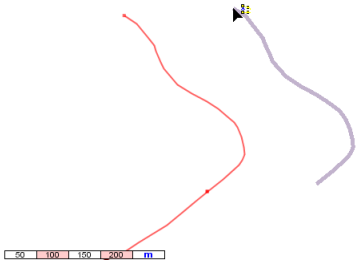
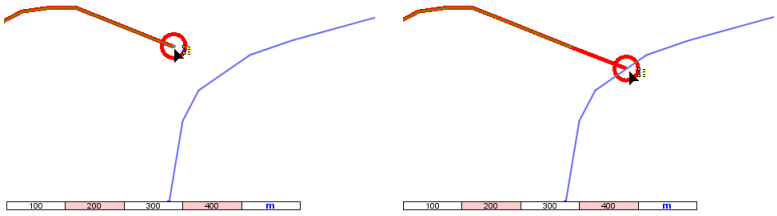


Figure 6-22 – The Shift Entire Segment adjustment method

Extend / Clip segment lets you extend (or contract) the selected segment along its current trajectory until it snaps to another segment. A split node will be created at the intersection to ensure correct topology. See the figure below:



(a) *(b)*

Figure 6-23 – (a) Before and (b) after extending the segment

Note: This and other tools that modify geometries are subject to **adjustment constraints** (if any). See *11.3* Adjustment Constraints for more information.

Note: To find inconsistent segments to adjust, use the **Query** window (see Section 10.2).

6.6 Auto-Connect Endpoint Tool

RoadMatcher uses purple discs to highlight nodes that cannot be separate, according to the topology of the networks in that area. Sometimes this is caused by incorrect matching; but if the matches are correct, often a simple remedy is to move related inconsistent nodes together using the **Auto Connect Endpoints Tool** (see below).

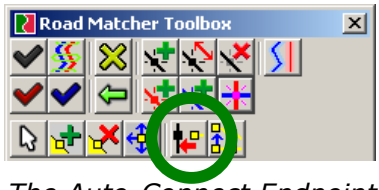


Figure 6-24 – The Auto-Connect Endpoint Tool (circled)

Simply select the **Auto Connect Endpoints Tool**, then click an inconsistent node. The other inconsistent nodes involved will be moved to the node you clicked on. This will often resolve the inconsistency.

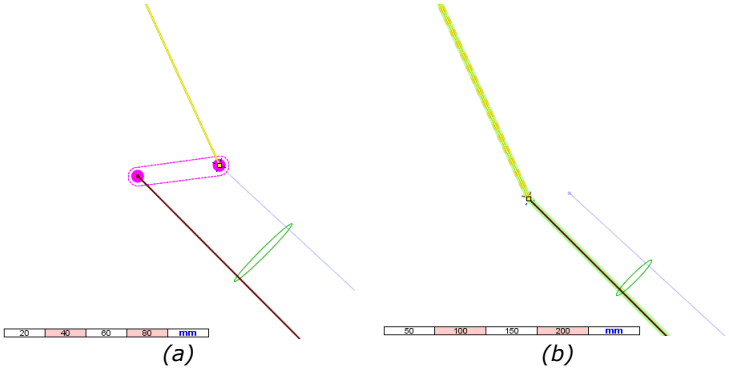


Figure 6-25 – (a) Before (b) After using the Auto-Connect Endpoint Tool. The purple discs have disappeared, and the segments are now green, indicating that the problem is resolved.

In the figure above, one of the segments is marked with a dashed yellow line. This indicates that the segment's geometry has been **adjusted** (modified) — in this case by only half a centimetre. (See *10.3* Legend to learn about the meanings of the various colours and symbols).

Note that a quick way to apply adjustments to all the data is AutoAdjustment (see *7.2* AutoAdjustment). But the Auto-Connect Endpoint Tool has a couple of useful features:

- It works when 2 **or more** related nodes are inconsistent (AutoAdjustment is limited to 2 related inconsistent nodes)
- It is not subject to the various AutoAdjustment tolerances

Typically you will run AutoAdjustment first, then decide whether to use the Auto-Connect Endpoint Tool on each inconsistency that remains.

To learn how to make more customized adjustments to the geometry, see *6.5* Adjust Endpoint Tool.

Note: This and other tools that modify geometries are subject to **adjustment constraints** (if any). See *11.3* Adjustment Constraints for more information.

Note: To find inconsistent segments to adjust, use the **Query** window (see Section 10.2).

Note: The adjustment method used by this tool is the one specified in the Adjustment Method section under the **AutoAdjust** tab in the **Conflation Options** dialog (see *11.4* AutoAdjust Options). This also specifies the adjustment method used for AutoAdjustment.

6.7 Vertex Editing Tools

You may come across a situation in which you need to adjust a segment, not at one of its endpoints but somewhere along the body of the segment. Examples of such adjustments are: moving an existing vertex, adding a new vertex and moving it, and deleting an existing vertex. RoadMatcher provides tools for these operations, shown in Figure 6-26 below:

Figure 6-26 – The Vertex Editing Tools (circled) – Select Features, Insert Vertex, Delete Vertex, Move Vertex.

Before using the **Insert Vertex**, **Delete Vertex**, and **Move Vertex Tools**, first select the segment using the **Select Features Tool**. Selecting a segment will show you what vertices are available (see Figure 6-27 below):

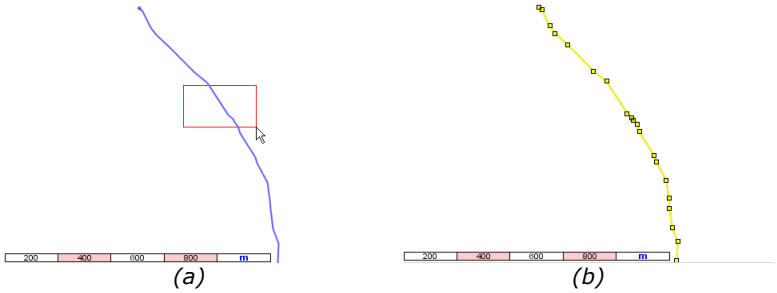


Figure 6-27 – (a) During and (b) after the use of the Selection Tool to show the existing vertices on a segment

Then, to insert a vertex, activate the **Insert Vertex Tool** and click on the desired location on the segment. To remove a vertex, activate the **Delete Vertex Tool** and click on the vertex. To move a vertex, activate the **Move Vertex Tool** and drag a vertex to a new location.

Note: These and other tools that modify geometries are subject to **adjustment constraints** (if any). See *11.3* Adjustment Constraints for more information.

6.8 Retire Tool

The **Retire Tool** is circled in the figure below:

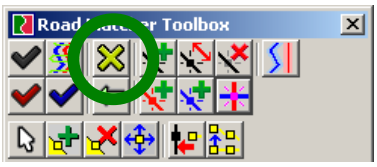


Figure 6-28 – The Retire Tool (circled)

Use the **Retire Tool** to indicate that a segment should not be included in the result road network. The Retired state is discussed in *3.2.1* Source States.

The figure below shows an example of matching a 4-way intersection to two 3-ways.

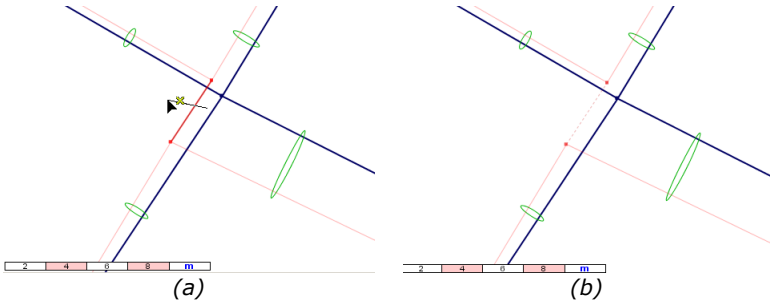


Figure 6-29 – (a) During and (b) after the use of the Retire Tool to retire the segment in the centre

In the example above, retiring the extra segment allows us to match the two networks despite their topologies being different.

If you attempt to retire a segment that has already been marked as Matched or Standalone, you will receive the following warning message:

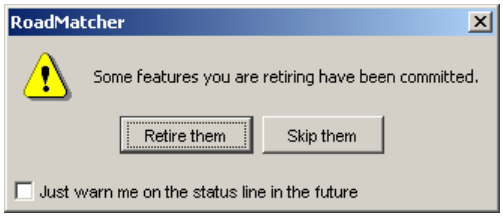


Figure 6-30 – Warning when retiring segments marked as Matched or Standalone

You may proceed with the retirement by clicking **Retire them,** or retire all but the Matched and Standalone segments by clicking **Skip them**. If you select the checkbox, the Retire Tool will always retire such segments, notifying you with a warning on the status line. If you then change your mind and want to be asked to

confirm every time, you can turn this dialog box back on – see *10.10* Application Options.

6.9 Undoing and Reverting

If you mistakenly retire a segment or incorrectly repair a topological inconsistency, the undo function (**Edit > Undo**) is a fail-safe. As in many applications, RoadMatcher's undo function remembers your actions sequentially and will undo those actions in reverse order.

Another approach is to use the **Revert Tool**. With this tool selected:

- if you drag across a segment whose geometry has been adjusted, it will change its geometry back to its original geometry
- if you drag across a segment whose geometry has **not** been adjusted, it will change its state back to the Unknown state (see *3.2.1 Source States*)

The **Revert Tool** is shown below:

Figure 6-31 – The Revert Tool (circled)

A more powerful alternative to the **Revert Tool** is the **Revert All In Fence** function. This function works on more than one segment:

- setting their states to Unknown
- returning them to their original geometries if they are adjusted
- deleting any split nodes encountered

To use it:

1. Activate the **Fence Tool** ().
2. Define an area by dragging the cursor to define a rectangle, or clicking several times to define a polygon (double-click to close off the polygon).
3. Click **RoadMatcher > Tools > Revert All In Fence**. All of the segments and split nodes found inside the area will be reverted as described above.

Note: An alternative way to use **Revert All** is to activate the **Revert Tool** then hold down **[Ctrl]**. You can then drag the cursor to define a rectangle -- all of the segments and split nodes found inside the rectangle will be reverted as described above.

There is a third kind of revert tool, the **Revert To Original Segment Tool** (). This tool causes segments to revert to their original state, by removing all split nodes. The tool is accessed by activating the **Revert Tool** then holding down **[Ctrl]**+**[Shift]+[Alt]**. To revert a segment:

1. Use the **Revert Tool** to ensure that all of the split sections of the segment are in the Unknown state.
2. Activate the **Revert To Original Segment Tool** by holding down **[Ctrl]**+**[Shift]+[Alt]**.
3. Drag the cursor across one of the split sections of the segment. All of the split sections of the segment will be replaced by the original, unsplit segment.

Note: This tool is primarily intended for use in exceptional circumstances (such as when an internal error is preventing you from using **Revert All** or the **Delete Split Node Tool**).

7 Automated Conflation

Automated conflation has been left until this section to give you an opportunity to first become familiar with the basic steps underlying roads conflation. Typically you will run AutoMatching and AutoAdjusting first, leaving the more complex cases to be resolved by manual conflation (Section 6).

7.1 AutoMatching

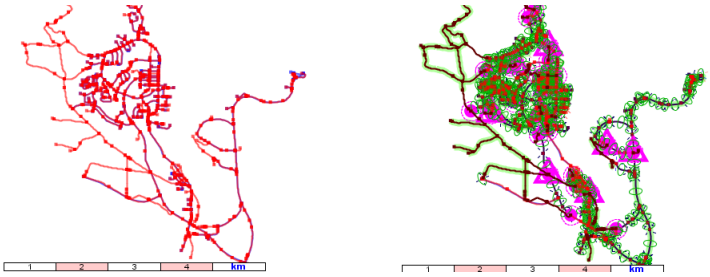
AutoMatching does the following:

- makes simple one-to-one matches between the two road datasets
- adds split nodes to resolve one-to-many and many-to-many matches.
- marks isolated segments as Standalone

An AutoMatch is run by clicking on **RoadMatcher > AutoConflate**, selecting the **AutoMatch** checkbox, then pressing **OK**.

Note: During AutoMatching, when establishing a match between two segments, RoadMatcher needs to know which to set as the reference and which the non-reference (see *3.2.1 Source States*). To view or edit this rule, see the instructions in *11.1* Match Precedence Rules.

The following figure shows the data before and after running an AutoMatch.



(a) *(b)*

Figure 7-1 – (a) Before and (b) after the AutoMatch operation

Segments highlighted in green are Integrated (see *3.2.2* Result States) – they have no problems and are ready to be exported as the conflation result (see *9* Exporting the Result). The green rings indicate matches. The purple markers indicate problems (see *10.3* Legend). Segments that are still red or blue are in the Unknown state – they were not processed by AutoMatch and will need to be handled manually (see *6* Manual Conflation).

For a summary of the state of conflation at this point, examine the **Statistics** display (see *10.1* Statistics).

The next step is to run an AutoAdjust to fix the simpler, more common problems – that process is described in Section 7.2.

7.2 AutoAdjustment

Earlier sections have discussed how to do manual adjustments of geometries (*6.5* Adjust Endpoint Tool*, 6.6* Auto-Connect Endpoint Tool). This section describes the AutoAdjust function, which performs controlled geometric adjustments on many segments, in order to resolve inconsistencies.

Like the **Auto-Connect Endpoint Tool** (Section 6.6), AutoAdjustment resolves inconsistencies by simply moving together related inconsistent nodes. But because it is applied to all the data, AutoAdjustment is subject to a number of restrictions for safety:

- It applies only to inconsistencies involving **two** nodes (the **Auto-Connect Endpoint Tool** works for two **or more** related inconsistent nodes)
- The two inconsistent nodes must not have any Unknown segments incident on them (see *3.2.1* Source States) i.e. the topological relationships must be fully known
- It is subject to various tolerances, which will be described below (the **Auto-Connect Endpoint Tool** is not)
- Like the **Auto-Connect Endpoint Tool** and other geometry-modifying tools, it is subject to **adjustment constraints** (Section 11.3)

Typically you would run AutoAdjustment after doing an AutoMatch (Section 7.1). AutoMatching usually results in a number of inconsistencies, many of which can be fixed using AutoAdjustment. If there are no inconsistencies according to the **Statistics** display (Section 10.1), there is no need to AutoAdjust the data.

Before you run an AutoAdjust, you may want to check that the adjustment constraints are set properly. For more information, see *11.3* Adjustment Constraints.

Note: Whenever you use one of the manual matching tools, RoadMatcher will AutoAdjust the segments to automatically fix simple inconsistencies. To turn this feature off, see *10.10* Application Options.

To commence AutoAdjustment, click **RoadMatcher > AutoConflate**, select the **AutoAdjust** checkbox, then press **OK**. RoadMatcher will then perform a few iterations of AutoAdjustment, which may take a few minutes. When it is complete, you will see a status report like the following:

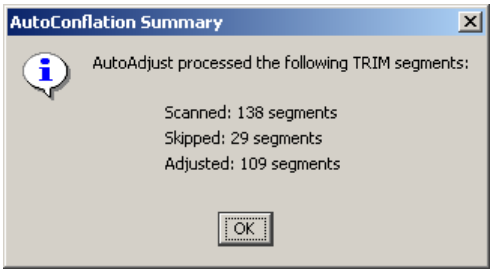


Figure 7-2 – AutoAdjustment report

In this example, of the 138 segments for which adjustment was to be attempted, 109 were actually adjusted. The remaining 29 were skipped for various reasons,

typically because of the safety restrictions mentioned earlier in this section. At this point you would find the remaining Inconsistent segments using the **Query** window (Section 10.2) and fix them using the manual adjustment tools (*6.5* Adjust Endpoint Tool*, 6.6* Auto-Connect Endpoint Tool).

If you are interested in visualizing the adjustments that were made, click **RoadMatcher > Tools > Generate Adjustment Vectors Layer** to see arrows showing the node displacements. This function works by comparing the current geometries to the original geometries.

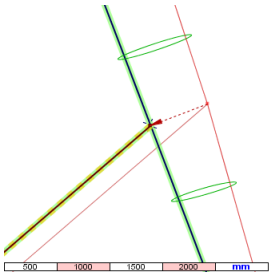


Figure 7-3 – An adjustment vector (dashed arrow). (The original dataset (TRIM) has been turned on to show the old geometry).

In Figure 7-3 above, the adjustment vector (dashed arrow) shows the adjustment for one segment – it has been shifted to the left by half a metre.

Note: To fix simple inconsistencies, an AutoAdjustment (see *7.2* AutoAdjustment) is performed whenever you use any of the manual matching tools (*6.1* Match/Commit Tool*, 6.2* Path Match Tool*, 6.3* Define Paths Tool). To turn this feature off, see *10.10* Application Options.

7.3 The Next Step

Once you have run an AutoMatch and AutoAdjust, the next steps are to (1) fix any inconsistencies that weren't fixed by the AutoAdjust (2) match any segments that weren't processed by the AutoMatch.

To find the inconsistencies that remain, click the Zoom To Inconsistent Segment button, circled below:

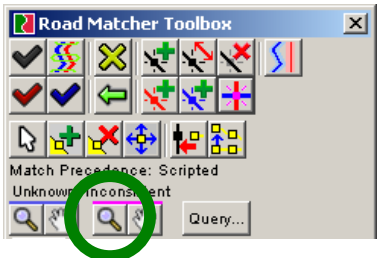


Figure 7-4 – The Zoom To Inconsistent Segment button (circled)

When you press the button, you will be taken to the nearest inconsistency, like the one shown below.

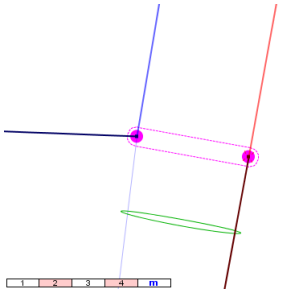


Figure 7-5 – An inconsistency left unfixed by the AutoAdjust operation

Because of the match (green ring), the included segments (dark blue and red) should be brought together. However, AutoAdjust left this situation unfixed because there are Unknown segments involved (bright blue and red). You would now use the manual conflation tools (Section 6) to resolve the situation: use the **Match/Commit Tool** to match the Unknown segments; then use the **Auto Connect Endpoint Tool** to bring the two nodes together.

Note: If you do not want to deal with this case now, you can leave it for later by right-clicking a purple disc and choosing Postpone Inconsistency. Then press the Zoom To Inconsistent Segment button to go to the next inconsistency. Later, you can reactivate postponed inconsistencies by clicking **RoadMatcher > Tools > Restore Postponed Inconsistencies**.

You will also want to match any segments that were left unprocessed by AutoMatch. To find them, press the Zoom To Unknown Segment button, shown below.

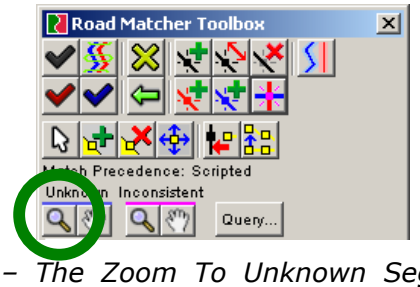


Figure 7-6 – The Zoom To Unknown Segment button (circled)

When you press this button, you will be taken to the closest segment in the Unknown state (see *3.2.1 Source States*). You can now use the **Match/Commit Tool** to match the segment or mark it as Standalone.

8 Saving the Conflation Session

You may save your conflation session at any time:

- to safeguard your work from data loss, should a critical system or application failure occur;
- to save your work at the end of a work-period, so that the session can be resumed at another time.

To save the current session, click **RoadMatcher > Save Session As**. You will be prompted for a filename with which to save the session.

Later, when you want to re-open your session, click **RoadMatcher > Open Session** and select the file.

Note: Sessions saved in one version of RoadMatcher may not be compatible with a newer or older version of RoadMatcher.

9 Exporting the Result

A conflation session is near its end when no segments are Unknown, Pending or Inconsistent – all segments are Integrated or Retired (or Matched (Non-Reference)). The remaining task is to generate the combined dataset – the conflation result.

9.1 Generating a Result Layer

You can see what the result will look like by generating a temporary result layer. Click **RoadMatcher > Result > Generate Layer**. You will see the following prompt:

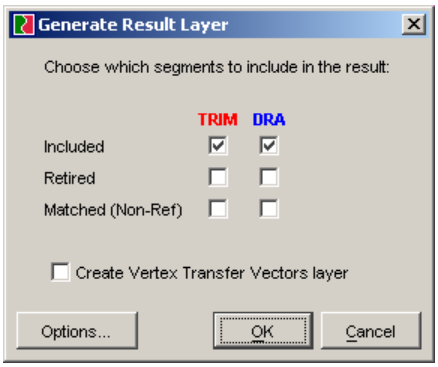


Figure 9-1 – The Generate Result Layer dialog.

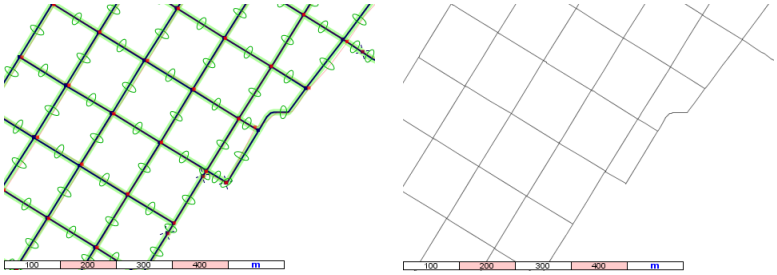
By default the result layer will contain only Included segments. Choose whether you would like to include Retired and Matched (Non-Reference) segments as well.

You also have the option to generate a Vertex Transfer layer. This is useful if you want to analyze the vertex transfers you specified earlier, if any (see 11.5.2 Vertex Transfer). Select this checkbox to generate a layer of arrows showing each vertex transfer.

The Options button takes you to the Result Options dialog (Section 11.5).

Press **OK** to generate the result layer (and optionally the Vertex Transfer layer).

The following figures show two source datasets for Fernie and the result dataset generated from them.



(a) (b)

Figure 9-2 – (a) The Fernie source datasets, successfully conflated (b) The result dataset, produced from the Integrated segments in the Fernie source datasets

Note: The **conflation attributes** will always be included in the result dataset. For descriptions of the conflation attributes, see *3.5* Conflation Attributes. In addition, there will be a "Source" attribute identifying which source dataset the geometry came from.

Note: RoadMatcher will warn you if there are any Unknown or Inconsistent segments when you generate the result layer. In particular, Unknown segments will be left out of the result layer.

9.2 Exporting a Result Package

The final step in the conflation process is to export a **result package**. This is a .zip file containing shapefiles for the following:

- the **result dataset** (whether or not you have explicitly generated it as described in *9.1 Generating a Result Layer*). Vertex transfers will be performed if you specified them (see *11.5.2 Vertex Transfer*). All attributes will be transferred, unless you have specified a subset of them as per *11.5.1 Source Attributes*.

- the **two source datasets**, including conflation attributes (Section 3.4). All roads are present in these datasets, including segments in the Retired and Unknown states. Any attributes that you have chosen to filter out (*11.5.1 Source Attributes*) will be excluded.

- the **issue log** if you have made any annotations using the issue log system (see *10.6* Logging Issues for more information).

To export a result package, click **RoadMatcher > Result > Export Package**, specify a filename, and hit **Save**.

The following shows the contents of a result package, using the popular WinZip program:

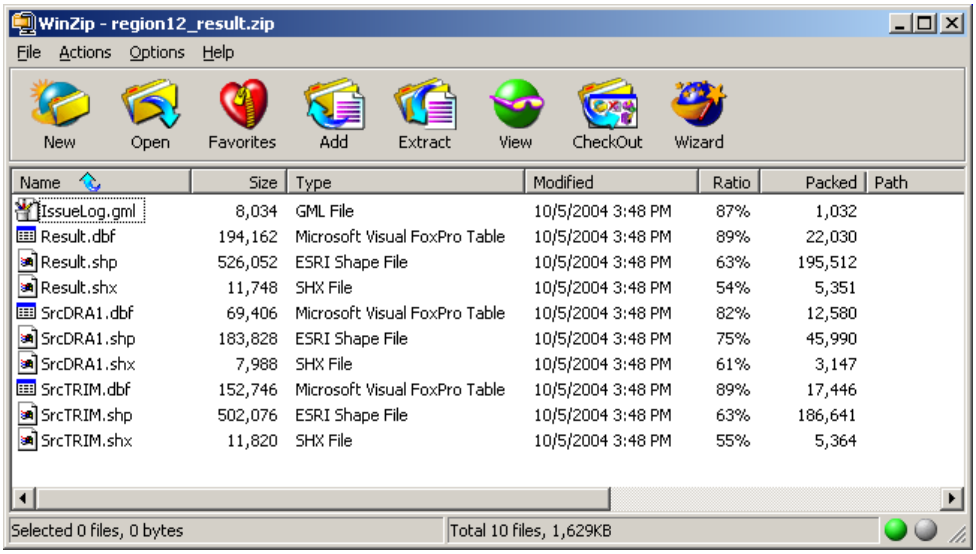


Figure 9-3 – Contents of the result package

Note: A result package is different from a source package (see *5.2 Creating a Session from* a Source Package). You cannot use the **New Session From Source Package** function to open a result package. To view the contents of a result package, unzip it to a directory of your choice, then open the shapefiles using **File > Load Dataset(s)**.

Note: The **conflation attributes** will be included in the exported source and result datasets. For descriptions of the conflation attributes, see *3.5* Conflation Attributes.

Note: RoadMatcher will warn you if there are any Unknown or Inconsistent segments when you export the result package. In particular, Unknown segments will be left out of the result dataset.

10 Other Tools

This section describes other useful RoadMatcher features.

10.1 Statistics

The **Stats** tab on the **RoadMatcher Toolbox** provides statistical information on the source datasets:

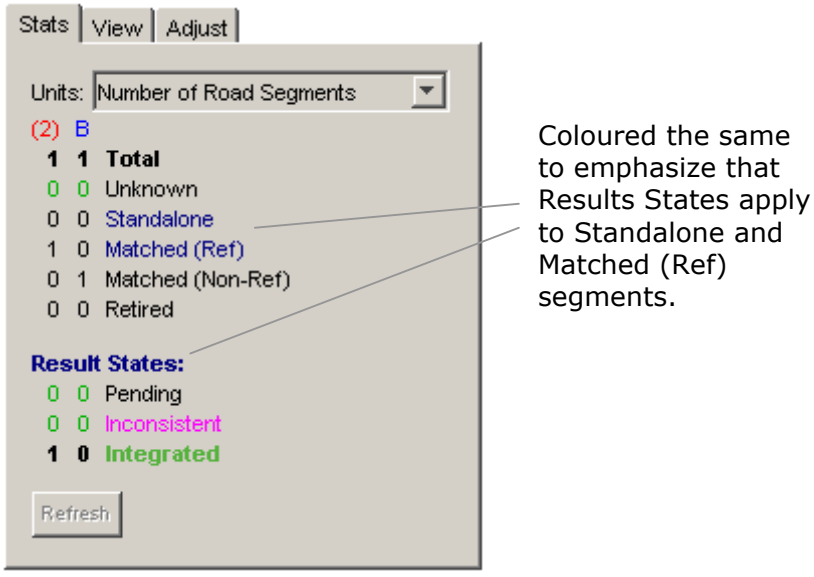


Figure 10-1 – The Stats tab on the RoadMatcher Toolbox

The **Units** drop-down enables you to select which set of statistics to view: number of segments, percentage of segments, length of segments or percentage of length of segments.

The first column in the tab displays the statistics for the first dataset; the second column, the second dataset. Statistics are broken down by segment state: Unknown, Standalone, Matched (Reference), Matched (Non-Reference) and Retired. The Standalone and Matched (Reference) segments are further broken down by result state: Pending, Inconsistent and Integrated. (For descriptions of each state, see *3.2* Segment States).

Although the statistics will automatically be updated as the matching session progresses, a **Refresh** button appears at the bottom of the **Toolbox** just in case.

As the conflation session progresses, the number of Unknown roads will decrease, eventually reaching zero when all Unknown roads have been processed. As the number of Unknown roads decreases, the other numbers will go up. The total of all roads in the Result States section will always be equal to the sum of Matched (Reference) and Standalone roads. As inconsistencies are resolved, the number of Inconsistent and Pending roads will decrease, while the number of Integrated roads will increase.

Alternatively, you can get the same information in HTML format by clicking **Tools > Statistics**:

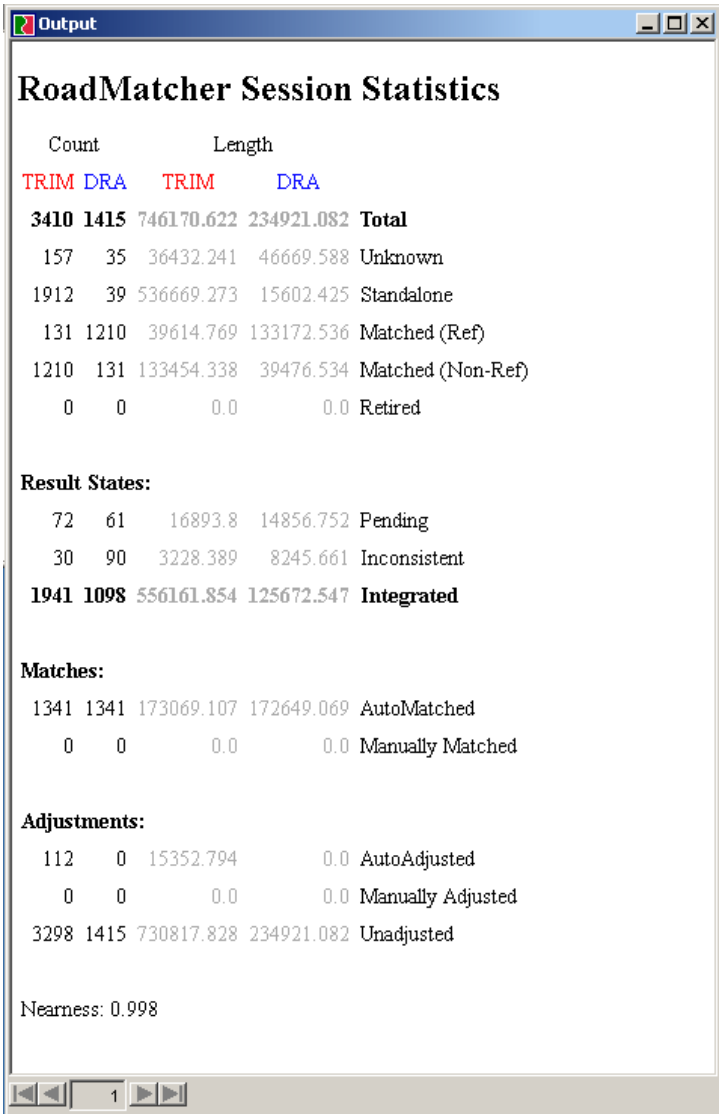


Figure 10-2 – The RoadMatcher Session Statistics report

Note that the report also includes AutoMatch and AutoAdjust statistics, as well as the average Nearness (weighted by segment length). (See *3.3.3* Nearness).

10.2 Query

The **Query** window enables you to search for segments based on certain criteria. It is especially useful for finding the Inconsistent or Unknown segments in each dataset. To open it, click **RoadMatcher > View > Query** (or click the **Query** button on the **RoadMatcher Toolbox**):

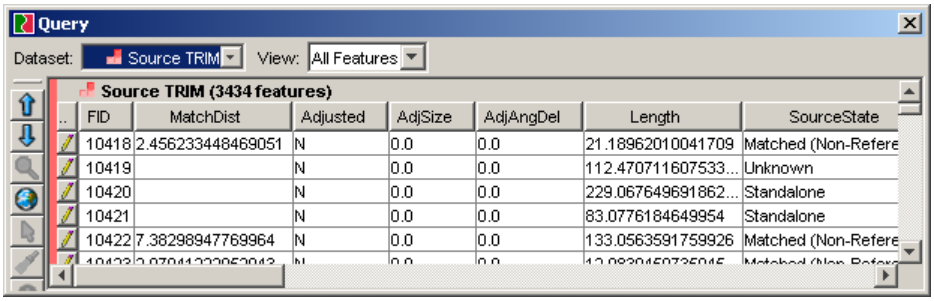


Figure 10-3 – The Query window

The **Query** window displays all of the attributes for one of the datasets, including the **conflation attributes** (Section 3.4). You can switch to the other dataset using the **Dataset** drop-down. To sort by one of the columns, click the column heading. To filter the information, click the **View** drop-down.

The buttons on the left of the **Query** window provide tools for interaction with the map:

Table 10-1 – Buttons of the Query window

	Zoom to Previous Row	Moves to the previous row of the table and displays that segment on the map.
	Zoom to Next Row	Moves to the next row of the table and displays that segment on the map.
	Zoom To Selected Rows	Zooms the map to the segments you have highlighted in the table.
	Zoom to Full Extent	Zooms out to display all the data
	Select In Task Window	Marks the segments highlighted in the table as selected on the map.
	Flash Selected Rows	Flashes on the map the segments highlighted in the table.
	Feature Info	Shows attribute information for the highlighted segment.

10.3 Legend

RoadMatcher assigns colours and symbols to segments for ease of identification. The **Legend** window displays these identifiers and their meanings – to display it, click **RoadMatcher > View > Legend** (see Figure 10-4 below).

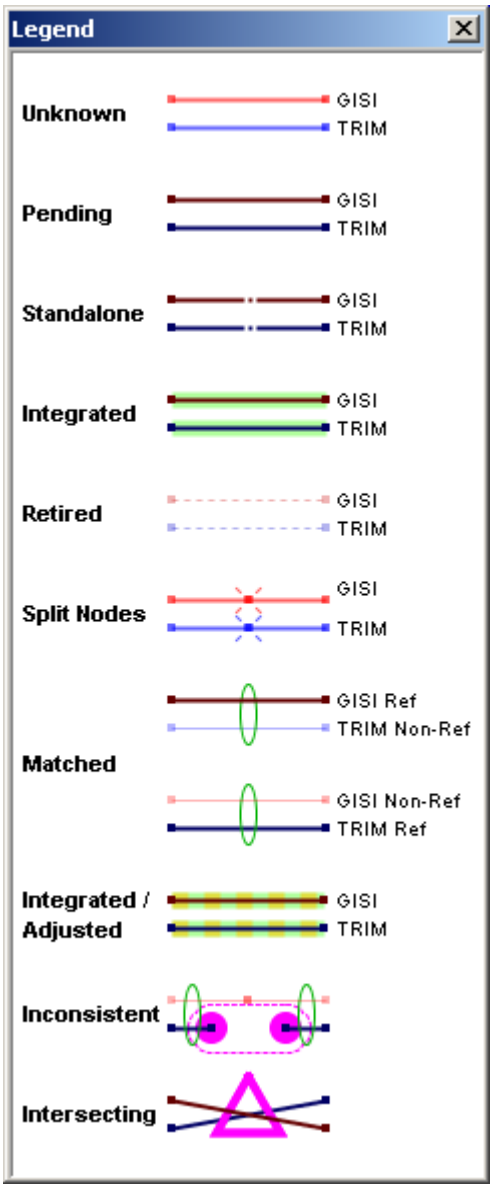


Figure 10-4 – The Legend window

For more information on Unknown, Pending, and the other states, see *3.2* Segment States.

10.4 Segment Comments

You can enter a comment for a segment – for example, to record why you decided to take a given action on the segment. Simply right-click the segment and select **Edit Segment Comment** from the pop-up menu. The **Edit Segment Comment** window will appear:

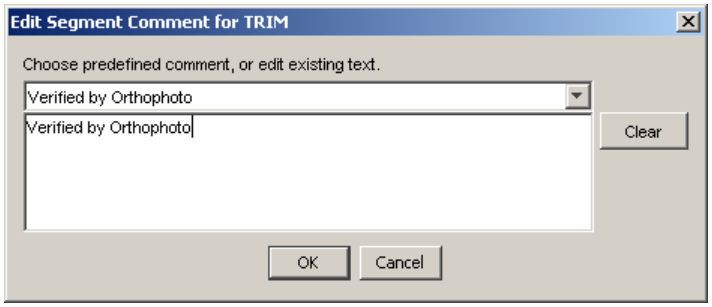


Figure 10-5 – The Edit Segment Comment window

Type in a comment, or choose a predefined one from the combobox (predefined comments are set in the profile – see the <issueLog> tag in *13* Appendix: Example of a Profile). Comments will become part of the result dataset (see *9* Exporting the Result).

You can see all of the comments in the **Query** window (Section 10.2).

If you cannot edit the comment text, the reason is that segment comments have been made uneditable. See the <segmentComments> tag in *13* Appendix: Example of a Profile.

Tip: To enter a comment for several segments at once, select them first then right-click one of them and click **Edit Segment Comment**.

10.5 Marking Segments As Reviewed

During your QA process, you may find it useful to have a way of identifying which segments have been QA'd. To do this, simply right-click a segment and select **Mark As Reviewed** from the pop-up menu. If you then open the **Query** window (Section 10.2) you will see that the Reviewed flag is now "Y" for that segment.

Similarly, you can use **Mark As Unreviewed** to set the Reviewed flag back to "N".

Tip: To mark several segments at once, select them first then right-click one of them and click **Mark As Reviewed** or **Mark As Unreviewed**.

10.6 Logging Issues

The **Issue Log** lets you mark a segment or area for future reference, typically to indicate questionable or anomalous situations. To create an issue for a segment, first select it using the **Select Features** tool. Then right-click it and select **Create/Edit Issue** from the pop-up menu. The following window will appear:

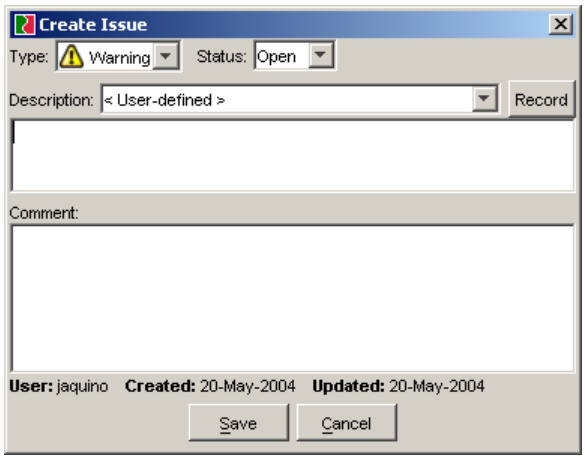


Figure 10-6 – The Create Issue window

Here you can type a description of the issue and enter comments about it. If you find that you are entering the same description frequently, store it by pressing **Record** – when creating similar issues in the future, you can get back the description clicking the **Description** drop-down.

If desired, you can click the **Type** drop-down to classify the issue as a Comment, Warning or Error. The **Status** drop-down lets you mark the issue as Open or Closed. Marking an issue as Closed is useful for retaining an issue for historical reasons, without highlighting it on the screen.

When you are finished, click **Save**.

Besides creating an issue for a segment, you can also create an issue for an arbitrary area, as follows:

4. Activate the **Fence Tool** (). You will use this tool to define the area.
5. Define the area by dragging the cursor to define a rectangle, or clicking several times to define a polygon (double-click to close off the polygon)
6. Right-click the area and select **Create/Edit Issue** from the pop-up menu. The **Create Issue** window will appear. Follow the above instructions on working with this window.

To indicate that an issue exists, the segment (or area) will be marked in orange, as shown below:

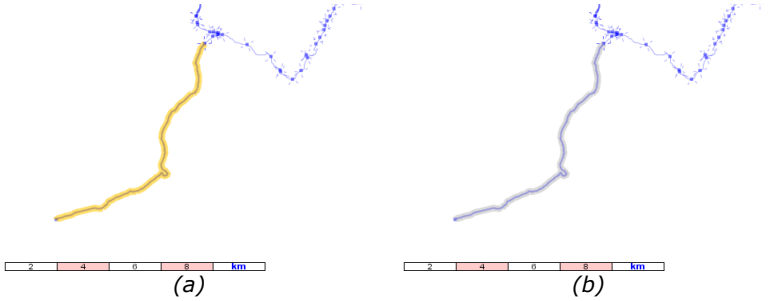


Figure 10-7 – (a) Orange marking indicating an open issue (b) Grey marking indicating a closed issue

When you see an issue indicated in orange, you can examine it by right-clicking it and selecting **Create/Edit Issue** from the pop-up menu. You will once again see the issue window (see Figure 10-6), from which you can make changes if desired.

Note: If you mark an issue as closed (see above), it will change from orange to grey.

To see a list of all issues, click **RoadMatcher > View > Issue Log**. The following window will appear:

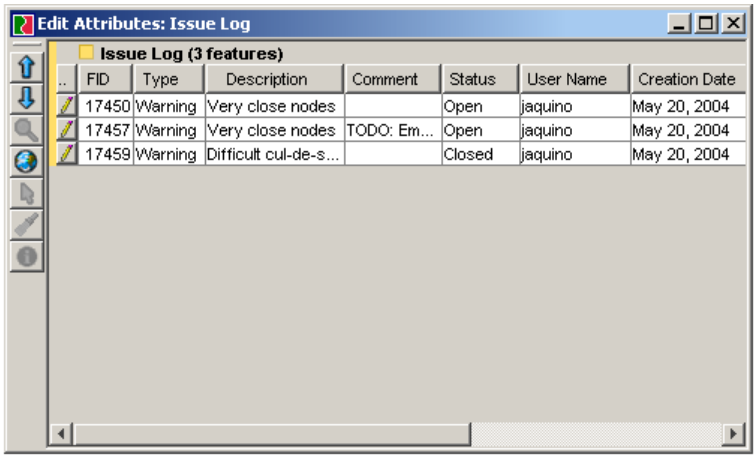


Figure 10-8 – The Issue Log window

To view the details of an issue, click the  icon – this will bring up the issue window (see Figure 10-6). To sort by Status, Type, or another column, click on the column heading. Use the down-arrow button and other buttons on the left to zoom to each issue – these buttons are the same as in the **Query** window and are described in Section 10.2.

Note: When you save your conflation session (Section 8), the issue log will be saved with it – next time you open the conflation session, the issue log will appear with it. Also, the issue log is included when you export a result package (Section 9.2).

10.7 Theming Layers

A theming layer enables you to highlight segments having a given attribute. Suppose you wanted to highlight in yellow all segments with a value of "paved" for the NAV_SURFAC attribute (assuming there is a NAV_SURFAC attribute). Here are the steps:

1. Click **RoadMatcher > Tools > Create Theming Layer**. You will be prompted to choose one of the two datasets:

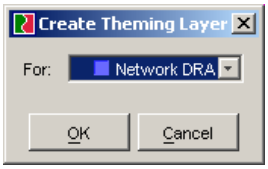


Figure 10-9 – Choosing a layer for which to add a theming layer.

2. Choose the dataset and press **OK**. The theming layer will be created.
3. Right-click the theming layer and click **Change Styles**.
4. Click the **Colour Theming** tab.
5. Select the **Enable Colour Theming** checkbox.
6. From the **Attribute** combobox, choose NAV_SURFAC. The different NAV_SURFAC values will appear in the table:

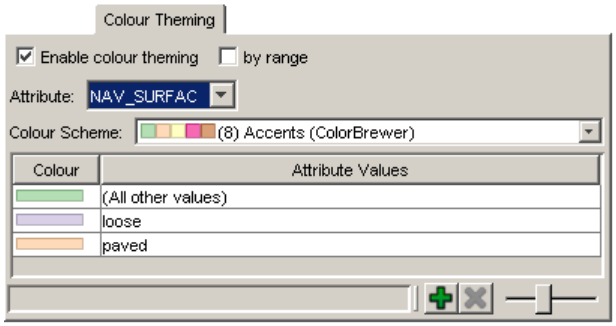


Figure 10-10 – The Colour Theming tab, showing the two NAV_SURFAC values: loose and paved.

7. Click the colour-bar to the left of "paved". A combobox will appear.
8. Select **Custom** (the first entry in the combobox). A dialog box will appear.

9. Specify a yellow line with a width of 12 pixels as follows:

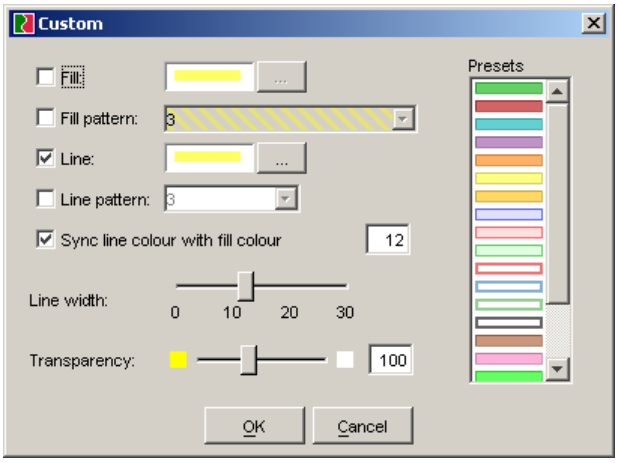


Figure 10-11 – Specifying a yellow line, 12 pixels wide, transparency = 100.

10. Press **OK** on the **Custom** dialog.
11. Press **OK** on the **Change Styles** dialog.
12. Paved roads will now be highlighted in yellow, as shown below:

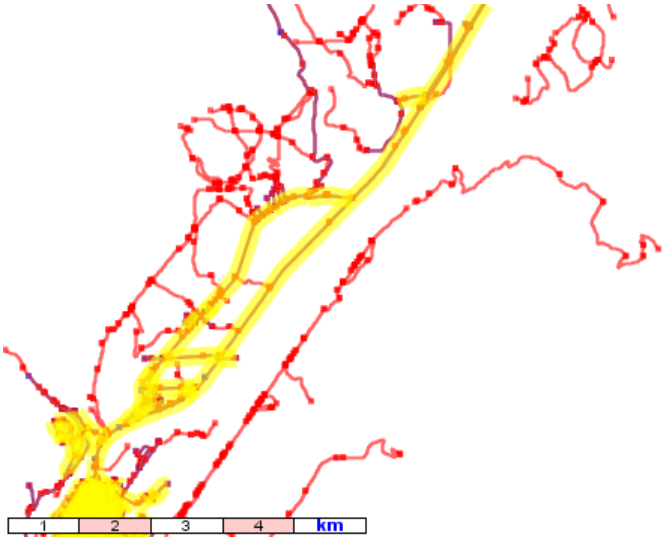


Figure 10-12 - Paved roads highlighted in yellow.

10.8 Finding Segments

Clicking **RoadMatcher > Tools > Find Segments** will display a list of special searches that you can choose from. Currently, there is only one search (**Find Overshoot Segments**), but more may be added in future versions of RoadMatcher.

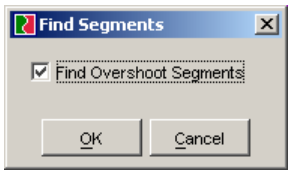


Figure 10-13 – The Find Segments window

Choosing **Find Overshoot Segments** will generate two layers: one for each of the two networks. Each layer will contain "overshoots" – segments connected to 0 segments at one end and 2 or more segments at the other end. If you right-click one of the Overshoots layers and select **View/Edit Attributes**, you will see that the segments are sorted from shortest to longest.

Note: Layers will not be generated if no overshoots are found.

10.9 View Options

The **View** tab on the **RoadMatcher Toolbox** gives you control over the colours and symbols appearing on the map:

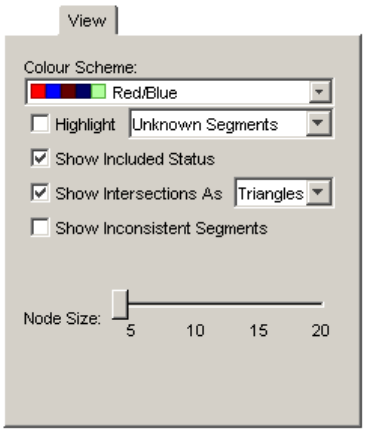


Figure 10-14 – The View tab on the RoadMatcher Toolbox

The **Colour Scheme** drop-down enables you to select an alternative colour scheme in which to work.

When **Highlight** is selected, RoadMatcher will mute all but the segments in the category selected from the adjacent drop-down: Unknown Segments, Included Segments or Inconsistent Segments. (See Section 3.2 for a description of the various segment states).

When **Show Result Status** is selected, RoadMatcher highlights in green the segments that have no problems and are ready to be exported as the result (see *9* Exporting the Result).

When **Show Intersections** is selected, intersections will be flagged on the map. You may choose the shape (triangles, squares, Xs or circles) from the drop-down.

When **Show Inconsistent Segments** is selected, RoadMatcher highlights in purple any segments whose topology is inconsistent with matching segments in the other dataset. (The default setting is off).

The **Node Size** slider lets you to increase or decrease the size of the split-node and end-node symbols.

Note: When you modify one of the above settings, the change is immediately reflected on the map.

Note: Changes you make to the view options described above will take effect for all conflation sessions, not just the current one. RoadMatcher cannot assign different view options (e.g. colour schemes) to different conflation sessions.

10.10 Application Options

This section describes miscellaneous RoadMatcher parameters that you can set. (RoadMatcher options pertaining to visualization are described in *10.9* View Options). To access the general RoadMatcher options, click **RoadMatcher > Application Options**:

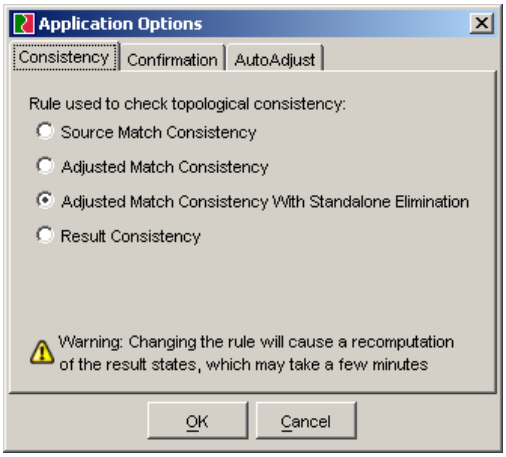


Figure 10-15 – Consistency options

Figure 10-15 above shows the **Consistency** options. Here you can select the rule used to determine whether the topology of one network is consistent with the topology of the other. It is highly recommended that you use the default value: **Adjusted Match Consistency With Standalone Elimination**. For more information on the various rules, see the *RoadMatcher Technical Design* document.

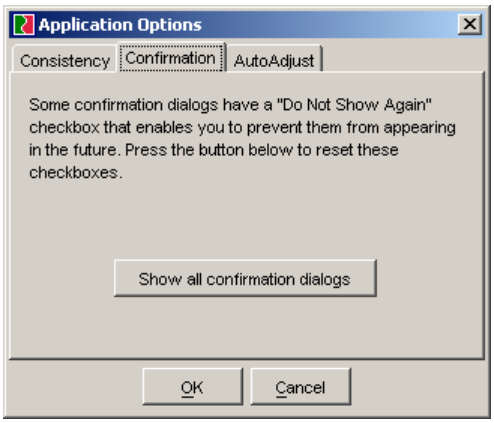


Figure 10-16 – Confirmation options

Figure 10-16 above shows the **Confirmation** options. By pressing the **Show all confirmation dialogs** button, you can turn back on all confirmation dialog boxes that you have turned off using the **Do not show again** checkbox at the bottom of those dialog boxes.

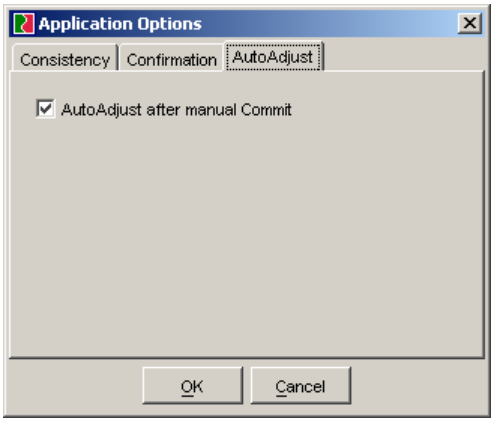


Figure 10-17 – AutoAdjust options

Figure 10-17 above shows the **AutoAdjust** options. (For more AutoAdjust options such as tolerances, see *11.4* AutoAdjust Options). To fix simple inconsistencies, RoadMatcher performs an AutoAdjustment (see Section 7.2) whenever you use one of the manual matching tools (*6.1* Match/Commit Tool*, 6.2* Path Match Tool*, 6.3* Define Paths Tool). Click the checkbox to turn this feature on or off.

11 Defining a Conflation Profile

If you have started a conflation session using **RoadMatcher > New Session**, you will recall choosing a profile as the first step (see 5.3 Creating a Session). This section is for the RoadMatcher administrator – it describes how to create a profile that people can use for future conflation sessions.

A profile contains the following information:

- match precedence rules
- AutoMatch options
- adjustment constraints
- AutoAdjust options
- result options
- short names for the datasets

Once you have set these parameters to your liking, you then save them as a conflation profile as described in *11.6* Saving the Profile.

11.1 Match Precedence Rules

When establishing a match between two segments, RoadMatcher needs to know which one to mark as Matched (Reference), and which to mark as Matched (Non-Reference) (see *3.2.1* Source States). To set these rules, click **RoadMatcher > Conflation Options > Precedence**:

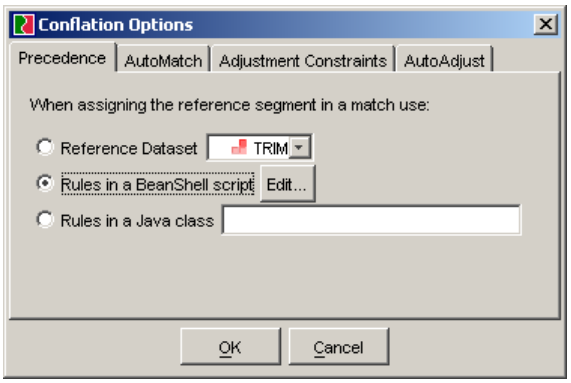


Figure 11-1 – Setting the Match Precedence Rules

Note: If all of the conflation options are greyed out, it is because the person who created the current conflation profile chose to "lock" it (see *11.6 Saving the Profile* on page 65).

The first option, **Reference Dataset**, is the simplest – RoadMatcher will take the segment from the specified dataset to be the reference.

The second option, **Rules in a BeanShell script**, is more advanced – it allows you to create custom rules using the BeanShell scripting language (similar to Java). To enter or modify a script, press the **Edit** button.

The third option, **Rules in a Java class**, also allows you to create custom rules, but using an actual Java class. To use this option, type in the fully qualified name of a Java class that implements the PrecedenceRuleEngine interface e.g. *com.vividsolutions.jcs.conflate.roads.model.ReferenceDatasetPrecedenceRuleEngine*.

Note: When making a match, you can override the match precedence rules by using the **dataset-specific Match/Commit Tools** described in Section 6.1.

11.2 AutoMatch Options

To set the AutoMatch options, click **RoadMatcher > Conflation Options > AutoMatch**:

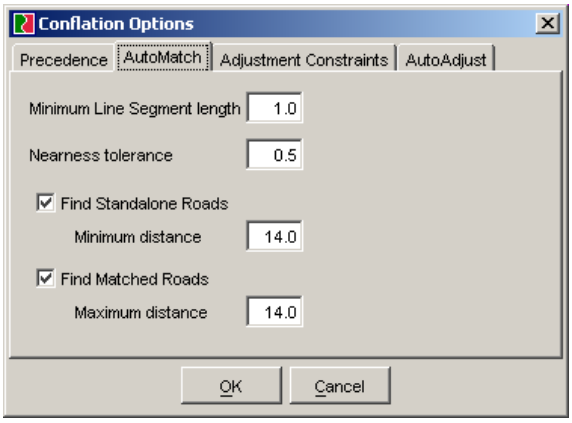


Figure 11-2 – AutoMatch options

Minimum Segment Length

The first field is the minimum **line-segment length** (note that this is vertex-vertex distance, not segment length). Whenever AutoMatch needs to split a segment to establish a match, it ensures that this tolerance is met.

Note: The manual adjustment tools also check this tolerance, warning you when it is not met.

Nearness Tolerance

The **nearness tolerance** defines what is meant by "near" in the calculation of nearness. The **nearness** is the fraction of the segment that is "near" to its matching segment. Nearness is one of the conflation attributes (see Section 3.4) – it is not used during AutoMatching; rather, people use it for QA purposes.

Find Standalone Roads

You may specify whether you want AutoMatch to mark isolated roads as Standalone (see *3.2.1* Source States). You must specify the **minimum** distance that a Standalone segment must be from another segment before it can be marked as Standalone. The higher the minimum distance value, the fewer the roads that will be marked as Standalone. The default value is 10 units.

Find Matched Roads

Finally, you may specify whether to enable or disable matching during an AutoMatch. You must specify the **maximum** distance that two segments may be from each other at any given point to be considered for matching. The lower the maximum distance value, the fewer the roads that will be matched. The default value is 10 units.

Note that the minimum and maximum distance parameters are approximate. Various heuristics are used to optimize processing times and to handle roads segments that interact with segments from the other dataset only at their endpoints.

Optimal AutoMatch settings for conflation will be different for different kinds of input datasets. Determining the appropriate AutoMatch settings will require experimentation with sample datasets.

11.3 Adjustment Constraints

An **adjustment constraint** is a safeguard against accidental modification of geometries. If a dataset has an adjustment constraint, you will be alerted if you try to modify the geometry of one of its segments.

To view or edit the adjustment constraints, go to **RoadMatcher > Conflation Options > Adjustment Constraints**:

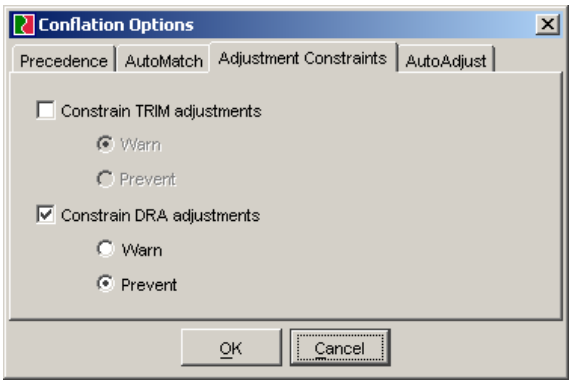


Figure 11-3 – Setting the adjustment constraints

Use the checkboxes to enable or disable adjustment constraints for each dataset. You can then choose whether to:

- allow geometries to be modified but be notified with a **warning**
- **prevent** geometries from being modified at all.

Typically adjustment constraints will be specified in the source package (see *5.2* Creating a Session from a Source Package) or profile (see *5.3* Creating a Session from Scratch).

Adjustment constraints apply to both manual adjustments (see the sections on Adjust Endpoint Tool, Auto-Connect Endpoint Tool, Vertex Editing Tools) and AutoAdjustment (Section 7.2).

Note: The creation and deletion of split nodes (see *6.4* Split Node Tools) are not considered to be adjustments, and thus are not subject to adjustment constraints.

11.4 AutoAdjust Options

Next, check the AutoAdjustment tolerances by clicking **RoadMatcher > Conflation Options > AutoAdjust**. You will see the following controls:

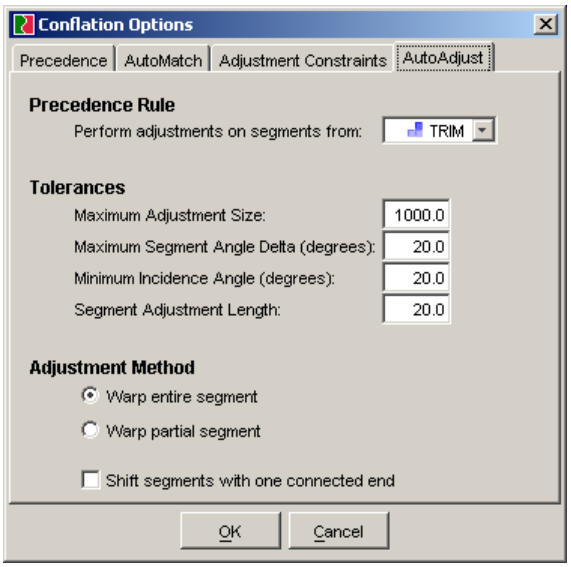


Figure 11-4 –AutoAdjust options

The first option specifies which dataset to modify.

The adjustment method used for AutoAdjustment is Warp Entire Segment (see *6.5* Adjust Endpoint Tool). For safety, AutoAdjust will perform only those adjustments that are within the following tolerances:

- **Maximum Adjustment Size** is the maximum node displacement for a segment.
- **Maximum Segment Angle Delta** is the maximum rotation of (1) the entire segment, if **Warp entire segment** is selected, or (2) the warped portion of the segment, if **Warp partial segment** is selected (see below)
- **Minimum Incidence Angle** is the minimum angle that the segment's nodes must make with the segments incident on them, after adjustment. This parameter prevents adjustments that create sharp angles between segments.
- **Segment Adjustment Length** specifies two things: (a) the length of segment that is warped when **Warp partial segment** is selected, and (b) half of the maximum segment length allowed by the **Shift short segments with one connected end** option (see below) i.e. segments longer than twice this length will not be shifted.

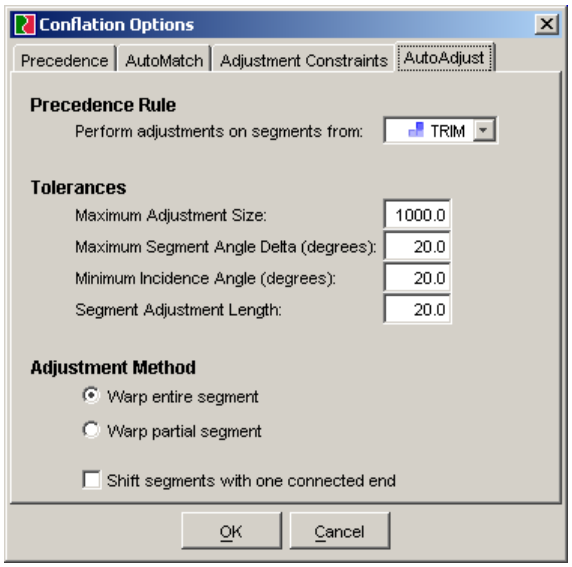


Figure 11-5 –AutoAdjust options

In addition, you have the following control over the way segments are adjusted:

- **Warp entire segment** will rotate and stretch the entire segment. This adjustment method ensures that no "kinks" are introduced into the segment, unlike **Warp partial segment**.

- **Warp partial segment** will rotate and stretch only the length of segment specified by the **Segment Adjustment Length** (see above). This adjustment method moves fewer vertices than **Warp entire segment**.

- **Shift short segments with one connected end** will simply move the entire segment if one end is "free" – no rotating or stretching. This is not applied to segments longer than twice the **Segment Adjustment Length** (see above).

Once you are satisfied with the AutoAdjustment parameters, press **OK**.

Note: The Adustment Method settings apply not only to AutoAdjustment but also to the Auto-Connect Endpoint Tool (see Section 6.6).

11.5 Result Options

This section describes how to set parameters pertaining to generating the result:

- Source Attributes
- Vertex Transfer

11.5.1 Source Attributes

By default, the result dataset will have all the attributes from both source datasets. A dataset with dozens of attributes can be unnecessarily large and difficult to work with. To select which source attributes to transfer into the result, click **RoadMatcher > Result > Options > Source Attributes**:

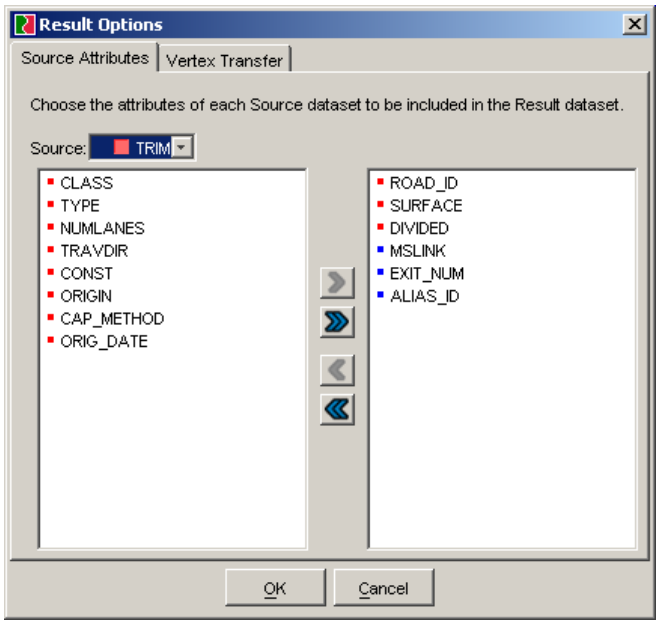


Figure 11-6 – Specifying the source attributes to go into the result.

Use the controls down the middle to shuffle attributes into and out of the result. To see the attributes for the other source dataset, click the **Source** drop-down. When you are finished, press **OK**. You will see your changes take effect when you generate a result layer (Section 9.1) or export a result package (Section 9.2).

Note: The **conflation attributes** will always be included in the result dataset. For descriptions of the conflation attributes, see *3.5* Conflation Attributes.

11.5.2 Vertex Transfer

When two segments are matched, the result geometry is taken from the *reference* segment. When vertex transfer is enabled, the *non-reference* segment also contributes spatial information, albeit in a limited sense. **Vertex transfer** is the mapping of vertices from the non-reference segment onto the result segment. This is done for two reasons:

- to preserve vertex density
- to preserve height information

Note: Vertex transfer does not change the source datasets – just the result dataset.

To set the vertex transfer options, click **RoadMatcher > Result > Options > Vertex Transfer**. You will see the following:

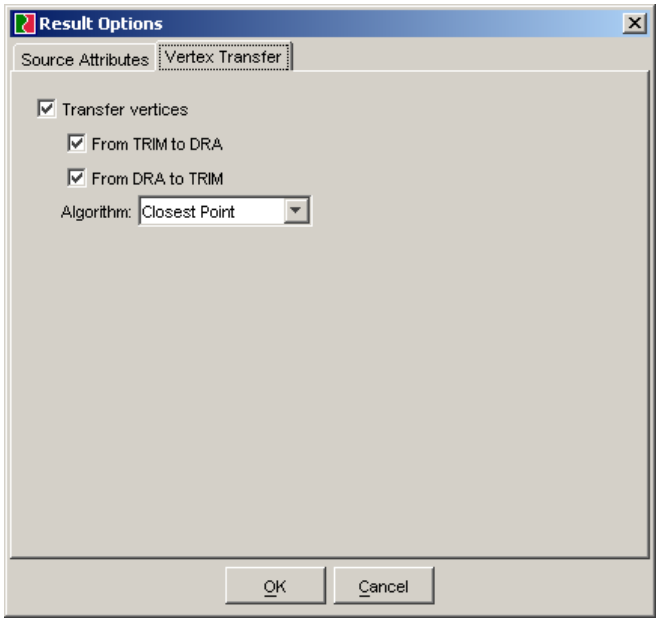


Figure 11-7 – Vertex-transfer options

Use the first checkbox to turn vertex transfer on or off.

Select the **From dataset A to dataset B** checkbox to transfer vertices from dataset A to dataset B when a non-reference segment from A is matched to a reference segment from B.

Select the **From dataset B to dataset A** checkbox to transfer vertices from dataset B to dataset A when a non-reference segment from B is matched to a reference segment from A.

The **Algorithm** drop-down allows you to specify the algorithm that will be used to compute the vertex transfers selected in the preceding checkboxes:

- **Closest Point** transfers a vertex to the closest point on the destination linestring. This technique is advantageous because it positions the destination vertex as close as possible to the source vertex. The technique is disadvantageous if the source and destination linestrings are offset by a large distance because some vertices at the ends of the linestring may be discarded.
- **Proportional Length** is based on the relative length of the vertex along the linestring. This technique is advantageous in that most or all of the source vertices will be correctly transferred. The technique is disadvantageous because the distance between source and destination vertices can sometimes be quite large.

11.6 Saving the Profile

Now that you have set the various parameters as described in the preceding sections, the next step is to create the profile by clicking **RoadMatcher > Save Profile As**.

You will first be asked whether you want to **lock** the profile – that is, whether to prevent users from changing the conflation parameters when they are working on a conflation session. This is a good idea in most cases.

Figure 11-8 – Locking the profile

You will then be asked for a filename with which to save the profile. The profile will be an XML file, so you can examine it with any text editor such as Notepad.

To set the short names of the two datasets, open the profile with a text editor and edit the "shortName" fields.

For an example of a profile, see Appendix: Example of a Profile.

12 Creating a Source Package

In Section 5.2, we discussed the easiest way to get a conflation session up and running: opening a source package. This section is for the RoadMatcher administrator – it describes how to create source packages.

A source package is a zip file that contains the files listed below.

Table 12-1 – Files in the source package

Name	Description
Manifest (manifest.xml)	A small XML file that specifies the names of the other files
Source Dataset 1	The datasets to be conflated
Source Dataset 2	
Conflation Profile (*.rmprofile)	(Optional) Predefined settings for various conflation parameters. See *11* Defining a Conflation Profile.
Node Constraints Dataset 1	(Optional) Point datasets specifying unmovable nodes for Source Datasets 1 & 2. See Figure 5-8 – Selecting node-constraint layers on page 21.
Node Constraints Dataset 2	
Context Dataset 1	(Optional) Read-only features surrounding Source Datasets 1 & 2
Context Dataset 2	

The various datasets are typically in Shapefile format, but may also be WKT, FME GML or JML.

Below is an example of a manifest. Note that optional fields can be left blank. Possible values for the format field are: shape, wkt, fmegml, jml.

```
<?xml version="1.0" encoding="UTF-8"?>
<sourceManifest>
  <dataset>
    <shortName>TRIM</shortName>
    <input>
      <fileName>fernie_trim.shp</fileName>
      <format>shape</format>
    </input>
    <nodeConstraints>
      <fileName></fileName>
      <format></format>
    </nodeConstraints>
    <context>
      <fileName></fileName>
      <format></format>
    </context>
  </dataset>
  <dataset>
    <shortName>DRA</shortName>
    <input>
```

```
    <fileName>fernie_dra.shp</fileName>
    <format>shape</format>
  </input>
  <nodeConstraints>
    <fileName></fileName>
    <format></format>
  </nodeConstraints>
  <context>
    <fileName></fileName>
    <format></format>
  </context>
 </dataset>
</sourceManifest>
```

Listing 12-1 – Example of a manifest file

Tip: You can leave out <nodeConstraints> or <context> tags if they are empty (since RoadMatcher 1.2).

13 Appendix: Example of a Profile

```
<?xml version="1.0" encoding="UTF-8"?>
<profile>
  <locked>true</locked>
  <dataset>
    <shortName>TRIM</shortName>
    <adjustmentConstraint>none</adjustmentConstraint>
    <resultAttributes>SURFACE, NUMLANES, TRAVDIR</resultAttributes>
    <vertexTransferFrom>false</vertexTransferFrom>
  </dataset>
  <dataset>
    <shortName>DRA</shortName>
    <adjustmentConstraint>prevent</adjustmentConstraint>
    <resultAttributes>NAV_RD_SCL, ALIAS_NAME, SEG_LENGTH</resultAttributes>
    <vertexTransferFrom>false</vertexTransferFrom>
  </dataset>
  <autoMatch>
    <minimumLineSegmentLength>1.0</minimumLineSegmentLength>
    <nearnessTolerance>10</nearnessTolerance>
    <findStandaloneRoads>
      <enabled>true</enabled>
      <minimumDistance>15.0</minimumDistance>
    </findStandaloneRoads>
    <findMatchedRoads>
      <enabled>true</enabled>
      <maximumDistance>15.0</maximumDistance>
    </findMatchedRoads>
  </autoMatch>
  <autoAdjust>
    <maximumAdjustmentSize>1000.0</maximumAdjustmentSize>
    <maximumSegmentAngleDelta>20.0</maximumSegmentAngleDelta>
    <minimumIncidenceAngle>20.0</minimumIncidenceAngle>
    <datasetName>TRIM</datasetName>
    <segmentAdjustmentLength>20.0</segmentAdjustmentLength>
    <methodClass>com.vividsolutions.jcs.plugin.conflate.roads.WarpAdjustmentMeth
od</methodClass>
    <shiftingSegmentsWithOneConnectedEnd>false</shiftingSegmentsWithOneConne
ctedEnd>
  </autoAdjust>
  <vertexTransfer>
    <enabled>true</enabled>
    <algorithm>com.vividsolutions.jcs.conflate.roads.vertextransfer.ClosestPointVerte
xTransferOp</algorithm>
  </vertexTransfer>
  <precedenceRuleEngine class="com.vividsolutions.jcs.conflate.roads.model.Scripte
dPrecedenceRuleEngine">
    <script>chooseReference(a, b) {
      trim = a.network.name.equals("TRIM") ? a : b;
```

```
      gisi = a.network.name.equals("TRIM") ? b : a;
      return gisi.feature.getString("POST_BY").trim().equalsIgnoreCase("GPS") ? gisi
 : trim;
    }</script>
 </precedenceRuleEngine>
 <issueLog>
  <description>Test description 1</description>
  <description>Test description 2</description>
  <description>Test description 3</description>
 </issueLog>
 <segmentComments editable="true">
  <segmentComment>Test segment comment 1</segmentComment >
  <segmentComment>Test segment comment 2</segmentComment >
  <segmentComment>Test segment comment 3</segmentComment >
 </segmentComments>
 <onSessionLoad>
 CreateThemingLayerPlugIn.createLabelledThemingLayer("DRA Post_By Labels", "DRA", "POST_BY",
     8, "ABOVE_LINE", session, layerManager, wc, new Value[]{
         new Value("TRIM", Color.RED),
         new Value("SKETCH", Color.BLACK) }).setVisible(true);
 CreateThemingLayerPlugIn.createThemingLayer("DRA Post_By Highlighting", "DRA", "POST_BY",
     12, "2,6", session, layerManager, wc, new Value[]{
         new Value("TRIM", new Color(116, 60, 28, 150)),
         new Value("SKETCH", new Color(0, 0, 0, 55)) }).setVisible(true);
 </onSessionLoad>
</profile>
```

Listing 13-1 – Example of a profile

The **onSessionLoad** tag contains a BeanShell script that is run when a session is opened (with New Session, New Session From Source Package, or Open Session). In the example above, the script creates two layers:

- ***DRA Post_By Labels***. This layer displays the text of the POST_BY field in the DRA dataset. The text will be in 8-point font and will be located above the segments (valid values are ABOVE_LINE, ON_LINE, and BELOW_LINE). "TRIM" will be displayed in red; "SKETCH" will be displayed in black. Visible=true indicates that the layer will be turned on.

- ***DRA Post_By Highlighting***. This layer highlights roads based on the contents of the POST_BY field. The highlighting line will be 12 pixels wide and will have a dashing pattern of 2 pixels on, 6 pixels off. When POST_BY is "TRIM", the line will be brown; when POST_BY is "SKETCH", the line will be grey. Visible=true indicates that the layer will be turned on.

Note: The **onSessionLoad** code is not run when the **autoconflate** or **exportresult** command-line utilities are run.

14 Appendix: Command-Line Utilities

RoadMatcher comes with two command-line utilities to facilitate the batch processing of the initial and final stages of conflation for a large number of datasets.

14.1 autoconflate

The first utility is **autoconflate** – it takes a source package (Section 5.2) as input, runs an AutoMatch and an AutoAdjust, then generates a session file (Section 8). For example,

```
autoconflate region12.zip
```

will generate a session file `region12.rms`.

Note: The system calling **autoconflate** should ensure that (1) the datasets are valid (see *5.1.4* QA of Source Datasets) and (2) the source package contains a profile (see *12* Creating a Source Package – AutoMatch and AutoAdjust will use the parameters from the profile).

Note: If the datasets are not valid (see *5.1.4* QA of Source Datasets), an error message will be displayed and the session file will **not** be generated.

The session file generated by **autoconflate** can be loaded into RoadMatcher to perform manual conflation of any remaining Unknown or Inconsistent segments (see *6* Manual Conflation).

14.2 exportresult

The **exportresult** utility facilitates automated processing of completed RoadMatcher sessions. It allows result packages (Section 9) to be generated from sessions. For example,

```
exportresult region12.rms region12_result.zip
```

will generate a result package `region12_result.zip`. If desired, the result packages can have manual or automated QA performed on them to ensure that conflation has been carried out correctly.

15 Warnings and Error Messages

This section describes the error and warning messages that you might encounter when using RoadMatcher.

One or more problem nodes have segments in Unknown state

Context	Auto Connect Endpoint Tool
Cause	This tool requires that all segments incident on the problem nodes (purple discs) be in a state other than Unknown. This is so that the tool knows whether to adjust it (Matched (Reference) and Standalone states) or not (Matched (Non-Reference) and Retired)
Action	Change the state of Unknown segments using the Match/Commit Tool or Retire Tool

The new segment has a line segment shorter (...) than the minimum defined in the AutoMatch options (...).

Context	Insert Vertex Tool, Adjust Endpoint Tool, Auto Connect Endpoint Tool, Create Split Node Tool, Path Match Tool
Cause	Whenever RoadMatcher is about to adjust a segment's geometry, it checks that adjacent vertices are separated by the minimum line-segment length. This warning is displayed if the check fails.
Action	(1) To proceed with the adjustment, click Adjust Anyway. You can find short segments later using Tools > QA > Validate Selected Layers. (2) To abort the adjustment, click Cancel.

Warning: line-segment length < min length (... < ...)

Context	Insert Vertex Tool, Adjust Endpoint Tool, Auto Connect Endpoint Tool, Create Split Node Tool, Path Match Tool
Cause	Whenever RoadMatcher is about to adjust a segment's geometry, it checks that adjacent vertices are separated by the minimum line-segment length. This warning is displayed if the check failed but the adjustment was made anyway.
Action	(1) To find short segments like the one that caused the check to fail, use Tools > QA > Validate Selected Layers. You can do this now or later. (2) If you do not want adjustments that produce short segments, click Undo to revert the adjustment. If from now on you want to be asked to confirm such adjustments, go to RoadMatcher > Options > Confirmation and click Show Confirmation Dialogs.

Length = 0. Cancelled.

Context	Insert Vertex Tool, Adjust Endpoint Tool, Auto Connect Endpoint Tool, Create Split Node Tool, Path Match Tool
Cause	Whenever RoadMatcher is about to adjust a segment's geometry, it checks that the segment length is greater than 0. If this check fails, the adjustment is cancelled.
Action	Try the adjustment again, ensuring that the segment's start- and end-points are separated. If the end-point is being snapped to the start-point,

you may be zoomed too far out -- use the Zoom Tool to zoom in.

No inconsistent nodes here

Context	Auto Connect Endpoint Tool, Dismiss Warning menu item
Cause	The tool cannot find an Inconsistent node (purple disc) at the point you clicked on.
Action	Try clicking the Inconsistent node again -- ensure that you are clicking inside the purple disc.

Discontiguous split segments were detected. See View > Log for details.
Warning: Split segments discontiguous - see View > Log for details.

Context	Revert All Tool
Cause	"Bad" split segments were found. This error should never occur.
Action	Contact the RoadMatcher team.

The segments you are matching have already been committed.
One of the segments you are matching has already been committed.

Context	Match/Commit Tool
Cause	One or both of the segments that you are matching have already been matched or marked as Standalone.
Action	(1) To proceed with the match, click Match Them Anyway. Old match partners, if any, will be marked as Unknown. (2) To abort the match, click Cancel.

Warning: The segments you matched had already been committed.
Warning: One of the segments you matched had already been committed.

Context	Match/Commit Tool
Cause	One or both of the segments that you matched had already been matched or marked as Standalone.
Action	(1) Old match partners, if any, have been marked as Unknown. If you want to keep the new match, no action required. (2) If you do not want to keep the new match, click Undo. If from now on you want to be asked to confirm matches of segments that have already been matched or marked as Standalone, go to RoadMatcher > Options > Confirmation and click Show Confirmation Dialogs.

One segment you are marking as Standalone is part of a match.
Some segments you are marking as Standalone are each part of a match.
The segment you are marking as Standalone is part of a match.
The segments you are marking as Standalone are each part of a match.

Context	Match/Commit Tool
Cause	Some of the segments that you are marking as Standalone are already marked as Matched.
Action	(1) To sever the matches and mark the segments as Standalone, click Mark As Standalone Anyway. The old match partners will be marked as Unknown. (2) To prevent the matches from being severed, click Cancel.

Warning: One segment you marked as Standalone had been part of a match.
Warning: Some segments you marked as Standalone had each been part of

a match.
Warning: The segment you marked as Standalone had been part of a match.
Warning: The segments you marked as Standalone had each been part of a match.

Context	Match/Commit Tool
Cause	Some of the segments that you marked as Standalone had already been marked as Matched.
Action	(1) The matches have been severed and the old match partners marked as Unknown. If you want to keep the segments in their new Standalone state, no action required. (2) If you did not want the matches to be severed, click Undo. If from now on you want to be asked to confirm Standalone markings that would sever matches, go to RoadMatcher > Options > Confirmation and click Show Confirmation Dialogs.

No source segments here
No ...segments here

Context	Match/Commit Tool, Retire Segment Tool, Revert Segment Tool, Pan To Other End menu item, Create Split Node Tool, Define Paths Tool
Cause	In the region you dragged across, the tool cannot find any segments (from the specified dataset, if you are using one of the dataset-specific Match/Commit Tools or Create Split Node Tools).
Action	Try dragging across or clicking the segment(s) again. If you are using a dataset-specific Match/Commit Tool or Create Split Node Tool, ensure that you are dragging across or clicking segments from the appropriate dataset.

... ... and ... ... features specified. To make a match, specify only 1 feature from each.

Context	Match/Commit Tool
Cause	To make a match, the tool requires one segment from each of the two datasets. This requirement was not met.
Action	Try dragging across the two segments again. If there are segments in the way, then instead of dragging, click one then click the other.

Can't create split node on adjusted segment

Context	Create Split Node Tool
Cause	The tool can create split nodes only on segments that have not yet been adjusted.
Action	If possible, create the split node on another, unadjusted segment. Or unadjust the segment using the Revert Tool.

Cannot find target feature. Double-click to clear paths.

Context	Define Paths Tool
Cause	The tool failed find a way to extend the current path to the segment you specified.
Action	(1) If you want to create a new path, double-click anywhere on the map to clear the existing paths. (2) If you want to extend the current path, try specifying a closer segment.

The paths must contain only Unknown segments

Context	Define Paths Tool
Cause	The tool found a way to extend the current path to the segment you specified; however, the new path would contain Matched, Standalone, or Retired segments. The Match Paths function requires paths to contain Unknown segments only.
Action	Find the Unknown segments using the Zoom and Pan tools, then mark them as Unknown using the Revert Tool.

Can't delete split node from adjusted segment

Context	Delete Split Node Tool
Cause	This tool can only create split nodes on segments that have not yet been adjusted.
Action	Unadjust the split segments using the Revert Tool. Delete the split node using the Delete Split Node Tool. If desired, adjust the split-less segment using the Adjust Endpoint Tool.

No ...split nodes here

Context	Delete Split Node Tool
Cause	The tool cannot find a split node ("X" symbol) at the point you clicked on.
Action	Try clicking the split node again -- ensure that you are clicking on the X.

Warning: ... Unknown/Inconsistent segments

Context	Export Result menu item
Cause	The Export Result function created a Result layer (from the Included segments), but found that some segments were Unknown or Inconsistent. Ideally, the Export Result function would be used when there were no Unknown or Inconsistent segments.
Action	Mark Unknown segments as Matched, Standalone, or Retired. Use the Adjust Endpoint Tool and Auto Connect Endpoint Tool to fix Inconsistent segments.

No segments match the given criteria

Context	Find Closest Segment button
Cause	The Zoom/Pan To Unknown/Inconsistent/Adjusted Segment function cannot find any segments.
Action	No action required.

The two input paths must contain only Unknown segments

Context	Match Paths menu item
Cause	One or both of the input paths contain Matched, Standalone, or Retired segments. The Match Paths function requires paths to contain Unknown segments only.
Action	Find the Unknown segments using the Zoom and Pan tools, then mark them as Unknown using the Revert Tool.

A path must be defined for each source network using the Define Paths Tool

Context	Match Paths menu item
Cause	The Match Paths function requires two paths to be defined, one for each dataset. But there are none (or only one).
Action	Use the Define Paths Tool to specify the two paths to match.

Path is invalid because segments have changed. Please clear the path and define it again.

Context	Match Paths menu item
Cause	Split nodes must not be created or deleted on segments in the path; otherwise the path becomes invalid.
Action	Clear the paths using RoadMatcher > Clear Paths, then recreate them using the Define Paths Tool.

The two layers must be different

Context	New Session menu item
Cause	The New Session dialog box requires the two chosen input datasets to be different.
Action	Click one of the drop-downs to choose a different dataset.

The two short-names must be different

Context	New Session menu item
Cause	The New Session dialog box requires the two short-names to be different.
Action	Edit the short names so that they are different. RoadMatcher uses short-names in several places to identify the datasets.

A session already exists. Creating a new session will erase all the work you have done in the current session.

Context	New Session menu item
Cause	The active window already has a RoadMatcher session started on it.
Action	(1) If you want to discard the RoadMatcher layers (only the two original layers will remain) and begin a new RoadMatcher session, click Proceed Anyway. (2) If you want to keep the existing RoadMatcher session but also start a new one, first click Cancel. Then click File > New Task, open two datasets using File > Load Datasets, and begin your new session using RoadMatcher > New Session.

At least 2 non-conflation layers must exist

Context	New Session menu item
Cause	RoadMatcher requires two input datasets.
Action	Load at least two datasets using File > Load Datasets.

One or both of the source networks contains validation errors - please review before continuing.

Context	New Session menu item
Cause	Coincident segments were detected in one or both of the datasets (see the User's Guide for more information on coincident segments).
Action	Review the errors by right-clicking the Coincident Segments layer and

selecting View Attributes.

No endpoints here

Context: Adjust Endpoint Tool

Cause: The tool cannot find the start or end of any segments at the point you began dragging on.

Action: Try dragging the endpoint again -- ensure that you click right on the endpoint.

Endpoints too far apart (required = ..., actual = ...)

Context: Path Match Tool

Cause: This tool requires that, on the two segments you specify, one or both pairs of endpoints be within the maximum distance for matched roads.

Action: (1) Increase the maximum distance by going to RoadMatcher > Options > AutoMatch > Find Matched Roads: Maximum Distance, or (2) adjust the endpoints to be closer using the Adjust Endpoint Tool.

Please specify two Unknown segments, one from each network

Context: Path Match Tool

Cause: This tool requires you to drag across two Unknown segments. This error can occur if, for example, you drag across two Matched segments, or more than two Unknown segments.

Action: Try dragging across the segments again -- ensure that there are only two, and that they are both in the Unknown state.

Couldn't create split node on other segment because it has been adjusted

Context: Adjust Endpoint Tool (Clip/Extend Segment adjustment method)

Cause: When you extend a segment to a second segment (or clip a segment back to a second segment), it will create a split node on the second segment. Like other tools however, this tool can create split nodes only on segments that have not yet been adjusted.

Action: Unadjust the second segment using the Revert Tool. Again try clipping/extending the first segment to the second segment -- the tool will split the second segment at the intersection point.

No Unknown ...segments here

Context: Path Match Tool

Cause: In the region you dragged across, the Path Match Tool cannot find Unknown segments.

Action: Try dragging across the two segments again (one from each dataset) -- ensure that they are both in the Unknown state.

Unable to determine path (PathMatchBuilder)
Unable to determine path (RoadPathZipper)

Context: Path Match Tool

Cause: For one reason or another, the Path Match Tool failed to create a path.

Action: Try dragging across a different pair of segments. Alternatively, try creating the paths using the Define Paths Tool, then matching them using

the Match Paths menu item.

No ... feature here
No ... or ... features here

Context	Match/Commit Tool (precise-match mode)
Cause	The tool could not find a segment at the point you clicked on.
Action	Try it again: click on the first segment, then click on the second segment -- be sure to click directly on each segment. Alternatively, drag across both segments at once.

Already retired

Context	Retire Segment Tool
Cause	The segment you are retiring has already been retired.
Action	No action required.

Some features you are retiring have been committed.

Context	Retire Segment Tool
Cause	One or more of the segments that you are retiring have already been matched or marked as Standalone.
Action	(1) To change their states from Matched or Standalone to Retired, click Retire Them . Old match partners, if any, will be marked as Unknown. (2) To retire only the segments that are not Matched or Standalone, click Skip Them.

Warning: Some features you retired had been committed

Context	Retire Segment Tool
Cause	One or more of the segments that you are retiring have already been matched or marked as Standalone.
Action	(1) Old match partners, if any, have been marked as Unknown. If you want to keep the segments in their new Retired state, no action is required. (2) If you did not want the Matched and Standalone segments to be retired, click Undo. If from now on you want to be asked to confirm the retirement of Matched and Standalone segments, go to RoadMatcher > Options > Confirmation and click Show Confirmation Dialogs.

Already in the Unknown state

Context	Revert Segment Tool
Cause	The segment you are marking as Unknown is already marked Unknown.
Action	No action required.

Fewer than ... ...segments here

Context	Path Match Tool
Cause	The tool requires two segments, one from each of the two datasets. This requirement was not met.
Action	Try dragging across the two segments again -- ensure that there is one from each dataset.

A conflation session must be created or opened

Context	All tools and several menu items
Cause	A RoadMatcher session has not been started for the current window.
Action	Start a new session using RoadMatcher > New Session.

Start-node is not at a candidate adjustment location.
End-node is not at a candidate adjustment location.

Context	Inconsistent result state
Cause	The Result Consistency Rules have determined that one of the endpoints is not at a valid location.
Action	Use the Adjust Endpoint Tool to move the endpoint to one of the valid locations indicated by the small black arrows. Note: We recommend that the Adjusted Match Consistency Rules be used instead of the Result Consistency Rules.

Start-node is inconsistent with other nodes
End-node is inconsistent with other nodes

Context	Inconsistent result state
Cause	A comparison of the topology of the two networks suggests that the nodes indicated should not be separate.
Action	(1) Bring the nodes together, automatically using the Auto Connect Endpoint Tool, or manually using the Adjust Endpoint Tool, or (2) ignore the warning by right-clicking one of the purple discs and selecting Dismiss Warning.

Intersects ...

Context	Inconsistent result state
Cause	Two segments cross.
Action	(1) Adjust the segments so that they do not cross, using the Adjust Endpoint Tool, or (2) split both segments at the intersection point, using the black Create Split Node Tool.

Start-node neighbour ... in Unknown state
End-node neighbour ... in Unknown state
Match's start-node neighbour ... in Unknown state
Match's end-node neighbour ... in Unknown state

Context	Inconsistent result state
Cause	The result state of this segment cannot be determined yet because the state of a neighbouring segment is still Unknown.
Action	Change the state of Unknown neighbouring segments using the Match/Commit Tool or Retire Tool