

MRF Mapping Tool Kit **For MicroStation V8/XM**

May 2009



© MRF Geosystems Corporation
Suite 200, 625 – 14th Street N.W.
Calgary, Alberta
CANADA T2N 2A1

Tel: (403) 216-5515
Fax: (403) 216-5518
Email: mrfsales@mrf.com
WWW: <http://www.mrf.com>

TABLE OF CONTENTS

MRF Clean User's Guide

1. INTRODUCTION	2
1.1 Multi-tolerance Data Cleaning	2
1.2 Elements Processed by MRF Clean.....	5
2. MRF CLEAN MODULES	6
2.1 MRF Stroke.....	6
2.2 MRF Simplify.....	6
2.3 MRF Short Element Processor	7
2.4 MRF Extend	8
2.5 MRF Intersect	11
2.6 MRF Duplicate Remover	14
2.7 MRF Join.....	15
2.8 MRF Conflate	17
2.9 MRF Dangle Processor.....	17
3. RUNNING MRF CLEAN	19
3.1 Installation.....	19
3.2 Main Dialogue Box	19
3.3 Running MRF Clean - Step-by-Step	32
4. BATCH PROCESS	35
5. SAMPLE RESULTS OF MRF CLEAN	35
6. WORKFLOW RECOMMENDATIONS.....	40
6.1 General Processing Tips.....	40

6.2 Workflow Used in the Combined Module41

6.3 Custom Workflow Creation.....42

7. LEARNING MRF CLEAN WITH EXAMPLE DESIGN FILES 46

MRF FLAG EDITOR USER'S GUIDE..... 47

MRF Clean 3D User's Guide

INTRODUCTION..... 55

ELEMENTS PROCESSED BY MRF CLEAN 3D..... 58

MRF CLEAN 3D WORK FLOW..... 59

 Reading Input Elements59

 Intersection59

 Clustering61

 Splitting.....62

 Duplicate Removal63

 Element Joining.....64

 Dangle Removal66

 Conflation66

 Output67

MRF CLEAN 3D GUI 67

 MRF Clean 3D Input Parameters67

MRF CLEAN 3D PARAMETERS 75

SAMPLE RESULTS OF MRF CLEAN 3D..... 82

RUNNING MRF CLEAN 3D INTERACTIVELY 85

BATCH PROCESS 89

ADVANCED TOPICS 89

 What to do when different features are on the same level.....89

 Error messages90

 Memory requirements90

 Efficiency90

 Trouble shooting92

OPERATING TIPS..... 93

 Flagging true intersections93

 Dangle removal.....94

LEARNING MRF CLEAN 3D WITH EXAMPLE DESIGN FILES..... 95

MRF Polygon User's Guide

INTRODUCTION..... 98

ELEMENTS PROCESSED BY MRF POLYGON..... 99

MRF POLYGON PARAMETERS 100

 MRF Polygon Parameters.....100

 MRF Polygon Parameters Input.....104

RUNNING MRF POLYGON..... 108

BATCH PROCESS 110

RECOMMENDED MRF POLYGON WORK FLOW..... 110

 Work Flow 1:.....111

 Work Flow 2:.....113

LEARNING MRF POLYGON WITH EXAMPLE DESIGN FILES..... 115

MRF Clean USER'S GUIDE

For MicroStation V8/XM

May 2009



© MRF Geosystems Corporation
Suite 200, 625 – 14th Street N.W.
Calgary, Alberta
CANADA T2N 2A1

Tel: (403) 216-5515
Fax: (403) 216-5518
Email: mrfsales@mrf.com
WWW: <http://www.mrf.com>

1. Introduction

MRF Clean is a multi-level, multi-tolerance two-dimensional data cleaning software package. It is written in MicroStation Development Language (MDL), with some COM components. MRF Clean cleans linear elements, unshared cells and circular arcs. Shapes, complex shapes, complex chains and curves can also be cleaned if first stroked to line string elements. You have the option of cleaning circular parametric arcs in their native form, or stroking them prior to processing.

MRF Clean can:

- Correct over-shooting and under-shooting
- Perform line simplification
- Remove duplicate points, lines, etc.
- Extend line segments to maintain bearing (direction)
- Ease construction of topologies by creating intersections in crossing line-work
- Remove dangling line-work, or flag it for manual review and editing
- Join lines, line strings and arcs to form complex elements
- Merge line-work with approximately the same geometry (conflation)
- Reduce design file size to save disk space and increase display speed

1.1 *Multi-tolerance Data Cleaning*

Most GIS applications require the use of many data sets, which are collected by public agencies and private organizations, frequently using different levels of positional accuracy. Also, in a map, data sets are commonly stored on different levels. Since many spatial features share common boundaries, such as a forest stand and a lake, these common boundaries must match even when they are stored on different levels.

1.1.1 *Case Study – Resource Management*

As an example of the problems inherent in single-tolerance data cleaning, consider the following. A forest company uses two types of maps: maps containing lakes, rivers and roads from the state or provincial government, and forest stand maps collected by the company. Assume that the topographic and forest stand maps have positional accuracy's of two and four meters respectively. Let us also assume that the two types of maps are stored on different levels in the design file. In cleaning these two types of maps, the Forest Company faces several challenges:

- If the two levels are cleaned separately, lakes and forest stands may overlap after cleaning even though they should share a common boundary.
- If the two levels are merged into one level and cleaned using a single tolerance of two meters, then digitizing errors such as over-shooting and under-shooting in the line-work of the forest stand map may not be corrected.

or

- If the merged level is cleaned using a single tolerance of four meters, then the line-work from the topographic map will be generalized, reducing its accuracy for future operations.

1.1.2 The MRF Solution: Multi-tolerance Data Cleaning

Using multi-tolerance data cleaning, points with a larger tolerance can be moved a greater distance to coincide with points which have a smaller tolerance. In this way, inconsistencies such as the overlap or gap between a lake boundary and a forest stand boundary can be corrected by moving the less accurate boundary - the forest stand in the example - to the more accurate boundary of the lake. Also, when input points have identical tolerances, the movement of points is performed in such a way that straight lines are kept straight - a desirable behavior for many GIS/LIS applications. Using tolerances based on known quality limits of the input data, users can be confident that the data is being corrected automatically, without compromising the integrity of mission-critical information.

1.1.3 Other MRF Clean features

MRF Clean allows you to maintain inter-level geometric consistency. The following is an application where inter-level geometric consistency is desirable.

In property mapping, parcel boundaries on one level may not be geometrically consistent with corner markers on another level. For example, a lot boundary on level 40 may not end at the center of a cell on level 20 (i.e., *Figure 1.1.1*).

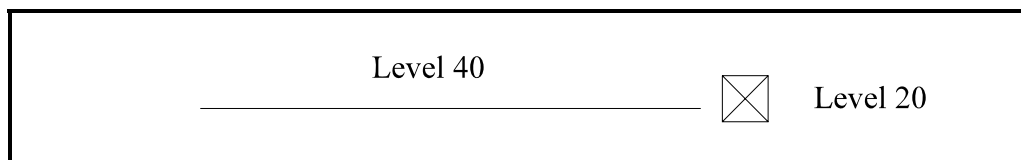


Figure 1.1.1. Example of geometric inconsistency

A user might assign the following tolerances to the listed entities:

ENTITY	TOLERANCE (mm)
Survey pins	1
Plan boundaries	3
Block boundaries	5
Lot boundaries	10

When cleaning up the property map, MRF Clean will move an entity with a larger tolerance to align with an entity with a smaller one:

- plan boundaries will end at accurately placed survey pins
- block boundaries will coincide exactly with plan boundaries
- lot boundaries will coincide exactly with block boundaries

In addition to the geometric advantages of multi-level, multi-tolerance cleaning, it is also important to note that by utilizing line-work on a number of levels, data redundancy can be substantially reduced. As an example, if an administrative boundary follows the bank of a river, the river's geometry can be used in data processing - eliminating the need to replicate that information to another level within the design file. This reduction in data duplication decreases the number of edits to be made when map updates are done and also reduces the size and complexity of the working design file.

MRF Clean is the result of many years of research and development. Extensive testing and user feedback indicate that MRF Clean can improve the quality of data conversion and data editing for mapping, GIS, and AM/FM applications. At the same time, **MRF Clean can increase productivity by two to ten times.**

1.2 Elements Processed by MRF Clean

MRF Clean can be used to process the following element types.

MSTN Element Type	MSTN Element Number	MRF Clean Processing
CELL_HEADER_ELM	2	Cleaned as a point feature ¹ or line strings ²
LINE_ELM	3	Converted to line
LINE_STRING_ELM	4	Converted to line string
SHAPE_ELM	6	Converted to line string
TEXT_NODE_ELM	7	Cleaned as a point feature
CURVE_ELM	11	Stroked to line strings
CMPLX_STRING_ELM	12	Stroked to line strings
TEXT_ELM	17	Cleaned as a point feature
CMPLX_SHAPE_ELM	14	Stroked to line strings
ARC_ELM ³	16	User Option: stroked to line strings or cleaned as an arc.
ELLIPSE_ELM	17	Converted to line string
ALL OTHER ELEMENTS ⁴		Copied to output ⁵

Note 1: When the cell is not a group hole, it is cleaned as a point feature. Cells can contain elements from multiple levels. The lowest of these levels is used when assigning tolerances etc.

Note 2: When the cell is a group hole, it is converted to a set of line strings. Each will be assigned the lowest level of the cell.

Note 3: Circular arcs (where primary axis and secondary axis are of equal length) can be processed in their native form by MRF Clean. You also have the option of stroking them before cleaning.

Note 4: Including non-circular arcs.

Note 5: If no output file is specified, these elements will be left in the input file.

2. MRF Clean Modules

The MRF Clean tool combined the following software modules in its Graphical User Interface. It is important for the users to understand how each module performs so that they can choose the right modules in the clean process.

2.1 MRF Stroke

MRF Stroke will stroke certain non-linear elements into line strings. The purpose of doing this is to reduce the complexity of the MRF Clean process. The elements that can be processed in this module include curves, shapes, ellipses, complex shapes, complex chains, group holes and circular arcs. You have the option of stroking circular arcs into line strings as well. Element stroking is the first step in MRF Clean.

2.1.1 Elements Processed by MRF Stroke

MSTN Element Type	MSTN type	MRF Stroke Processing
GROUP_HOLE_ELM	2	Candidate for stroking
SHAPE_ELM	6	Candidate for stroking
CURVE_ELM	11	Candidate for stroking
CMPLX_STRING_ELM	12	Candidate for stroking
CMPLX_SHAPE_ELM	14	Candidate for stroking
ELLIPSE_ELM	15	Candidate for stroking
ARC_ELM	16	Candidate for stroking
OTHER ELEMENTS ¹		Copied or not touched

Note 1: If no output file is specified, these elements will be left in the input file and will not be affected by the stroking process.

2.2 MRF Simplify

Line simplification (also sometimes referred to as line weeding or line generalization) involves the removal of line string vertices based on a specified tolerance.

Only line string elements are processed by MRF Simplify (if required, non-linear elements can be pre-stroked into line strings, using MRF Stroke (described above)). This is because line strings can be simplified much more efficiently than elements such as curves and arcs.

Figure 2.2.1 is a simple example that demonstrates the transformation of an element by the simplifying process. The result of MRF Simplify will depend on user-specified tolerances. The smaller the tolerance, the smaller the difference will be between the original line string and the simplified one.

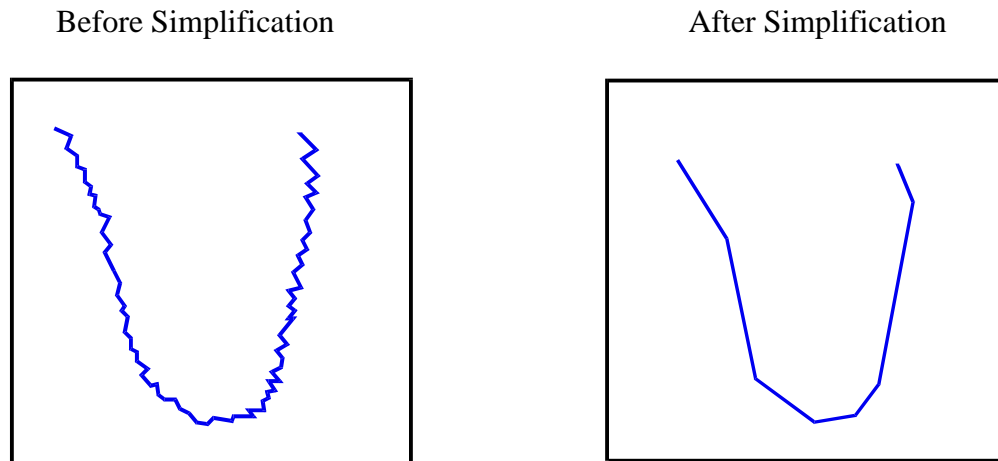


Figure 2.2.1. Line string before and after simplification

2.2.1 Elements Processed by MRF Simplify

MSTN Element Type	MSTN Element Number	Simplify Processing
LINE_STRING_ELM	4	Candidates for simplifying
ALL OTHER ELEMENTS		Copied to output ¹

Note 1: If no output file is specified, these elements will be left in the input file and will not be affected by the simplifying process.

2.3 MRF Short Element Processor

MRF Short Element Processor is used to flag or delete elements that have lengths smaller than the specified tolerances.

2.3.1 Elements Processed by MRF Short Element Processor

MSTN Element Type	MSTN Element Number	MRF Short Processing
LINE_ELM	3	Candidates for short element
LINE_STRING_ELM	4	Candidates for short element
ARC_ELM ¹	16	Candidates for short element
ALL OTHER ELEMENTS ²		Copied to output ³

Note 1: Both elliptical and circular (where primary axis and secondary axis are of equal length) arcs are processed by MRF Short Element Processor.

Note 2: Includes cells.

Note 3: If no output file is specified, these elements will be left in the input file and will not be affected by the short element process.

2.4 MRF Extend

MRF Extend is used to extend certain elements - correcting for undershoot - while maintaining line-work direction. This capability is critical to applications such as parcel mapping. If an element has a free end, MRF Extend will attempt to extend it until it meets other line-work within its tolerance. No intersections are created; element end-points are just extended.

MRF Extend is tolerance-controlled. If the extension distance is greater than the specified tolerance, the element will not be extended. The extension process is applied to line, line string and circular arc elements.

Elements are processed by MRF Extend in the following manner:

1. Line-line extension:

- a) In *Figure 2.4.1*, lines AB and CD will be extended to point E, if E is within tolerance of both AB and CD, and both B and D are free ends.

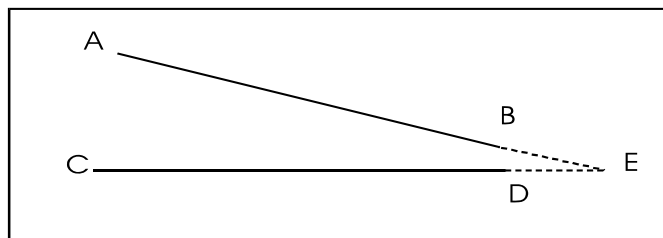


Figure 2.4.1. Linear extension example 1

- b) In *Figure 2.4.2*, lines AB and DE cannot be extended to point F, even though the distance BF is less than the tolerance for AB and EF is less than the tolerance for DE. This is because B is not a free-end.

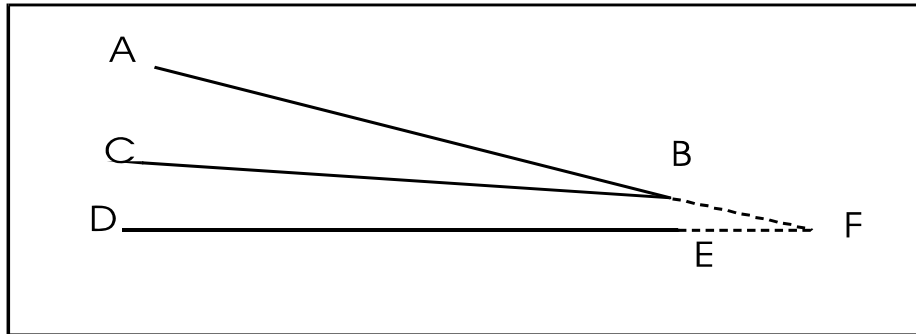


Figure 2.4.2. Linear extension example 2

- c) In *Figure 2.4.3*, CD has a tolerance larger than both the distances DE and CE. In this case, point D (as opposed to point C) will be extended to point E, since it has the smaller extension distance.



Figure 2.4.3. Linear extension example 3

- d) If a line can be extended to more than one element, it will be extended only as far as the closest one. In *Figure 2.4.4*, line AB will be extended to point C (not D or E).

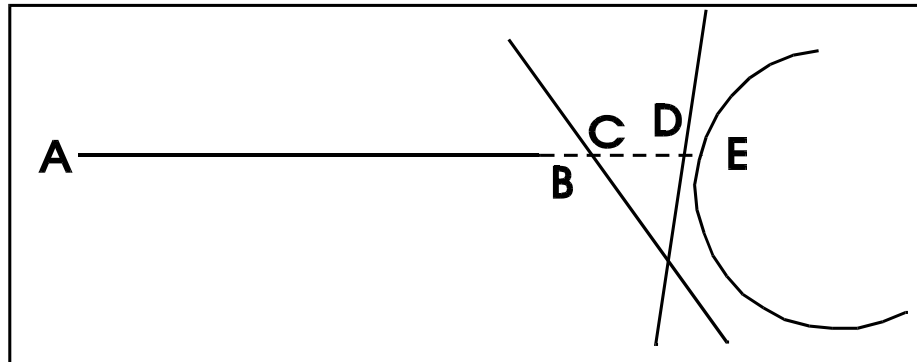


Figure 2.4.4. Linear extension example 4

2. **Line-arc extension:**

As shown in *Figure 2.4.5*, circular arc AB will be extended along its path to point C, if BC is less than the tolerance for AB. Also, line DE will be extended to DEF if EF is less than DE's tolerance.

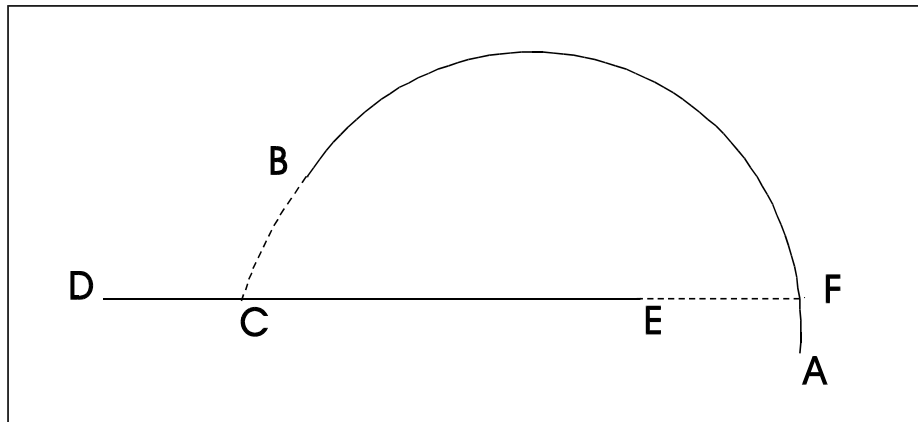


Figure 2.4.5. Line-arc extension example 5

3. **Arc-arc extension:**

As shown in the *Figure 2.4.6*, for arcs AB and DE, if BC is less than the tolerance for AB, then arc AB will be extended to C. Arc DE will be extended to F, if EF is less than the tolerance of DE.

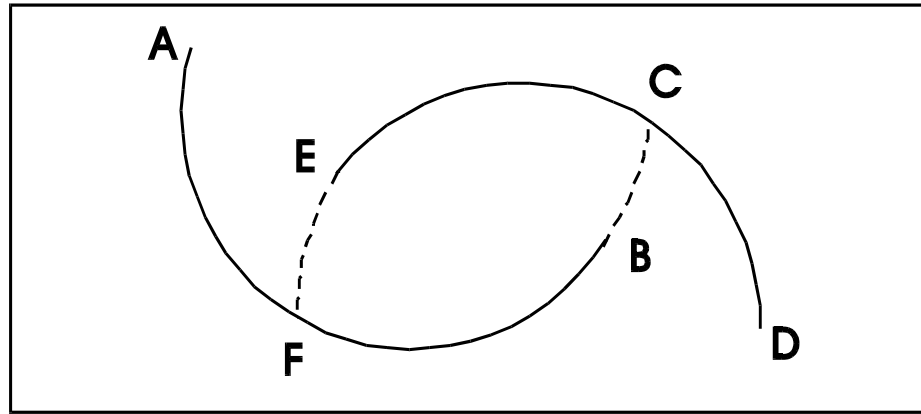


Figure 2.4.6. Arc-arc extension example 6

MRF Extend does not process overshoots. The Intersection and Dangle Processor modules, in combination, can serve this function.

2.4.1 Elements Processed by MRF Extend

MSTN Element Type	MSTN Element Number	Extend Processing
LINE_ELM	3	Candidates for extending
LINE_STRING_ELM	4	Candidates for extending
ARC_ELM ¹	16	Candidates for extending
ALL OTHER ELEMENTS ²		Copied to output ³

Note 1: Only arcs that are circular (primary axis and secondary axis are of equal length) are processed by MRF Extend.

Note 2: Includes non-circular arcs.

Note 3: If no output file is specified, these elements will be left in the input file.

2.5 MRF Intersect

MRF Intersect flags and/or creates true and fuzzy intersections. It has element intersection, clustering, and splitting sub-functions.

2.5.1 Intersection Creation

MRF Intersect recognizes two types of relationships:

True intersections: When two vectors cross each other, the crossover point (X) is called a true intersection (*Figure 2.5.1*)

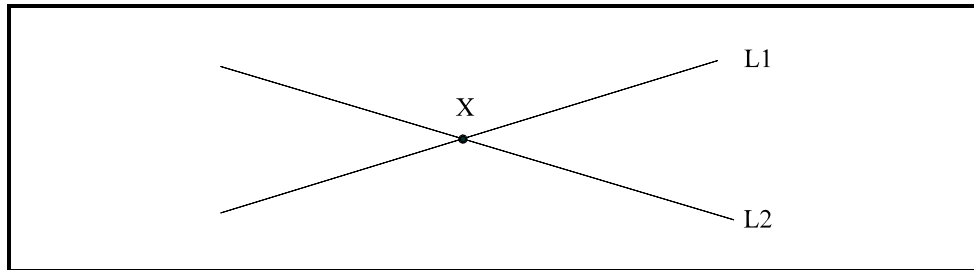


Figure 2.5.1. True intersection example

Fuzzy intersections: A fuzzy intersection can be created from elements which are within one of the tolerance distances, but do not actually touch or cross. In other words, the two lines nearly intersect, and the minimum distance from the end of one line to the other line is shorter than one of the lines' tolerances (*Figure 2.5.2*).

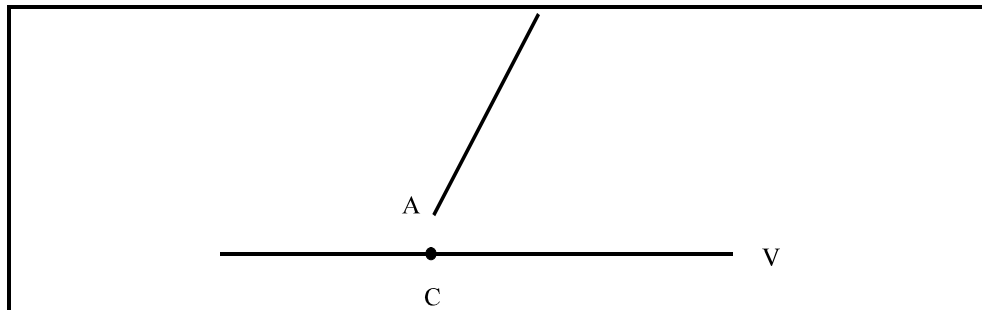


Figure 2.5.2. Fuzzy intersection example

If the distance from end point A to line V is shorter than one of the two lines' tolerances, then node A is projected the shortest distance to line V, to determine where the intersection is located. This projected point, point C, is called a **critical intersection** while A is called an **end intersection**.

Each intersection point inherits its tolerance from its input element. The End Intersection point A will have the same tolerance as the line that contains A and Critical Intersection C will have the same tolerance as that of line V.

2.5.2 Clustering

In this process, intersections are clustered together, or more specifically, intersections with larger tolerances are moved to intersections with smaller tolerances. However, no intersection is moved out of its tolerance. When points have identical tolerances, a true intersection has a higher priority than a critical intersection, which, in turn, has a higher priority than an end intersection. *Figure 2.5.3* shows the result of clustering point C to point A when line V has a larger tolerance than point A. On the other hand, if Line V has a tolerance smaller than that of point A, A will be clustered to point C (*Figure 2.5.4*).

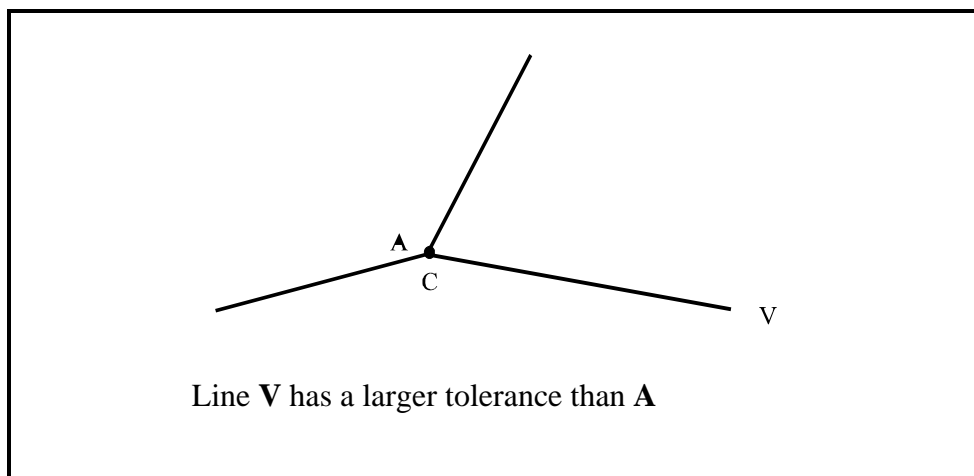


Figure 2.5.3. Intersection movement in clustering, case 1.

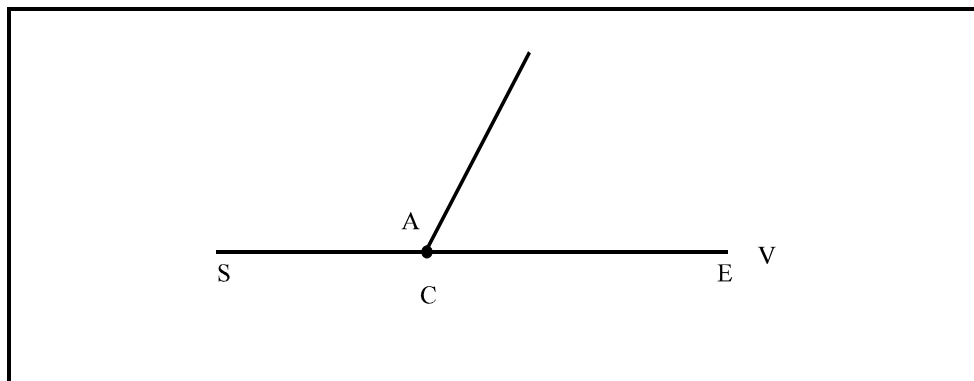


Figure 2.5.4. Intersection movement in clustering, case 2.

2.5.3 Splitting

Splitting causes input line strings to be split at their intersections to form a set of new line strings. For example, input line string V in *Figure 2.5.4* will be split into two line strings: SC and CE (assuming S and E are the start and end points of V). Each new

line string inherits the following attribute information from its corresponding input element:

Level	Property bits
Zhigh	Symbology
Zlow	(color, line weight & line style)
Graphic group number	Attribute linkages
Class	

After intersecting, clustering and element splitting, other modules in MRF Clean can be used to remove duplicates, and to join and conflate elements.

2.5.4 Elements Processed by MRF Intersect

MSTN Element Type	MSTN Element Number	MRF Intersect Processing
CELL_HEADER_ELM	2	Candidates for intersecting
LINE_ELM	3	Candidates for intersecting
LINE_STRING_ELM	4	Candidates for intersecting
ARC_ELM ¹	16	Candidates for intersecting
ALL OTHER ELEMENTS ²		Copied to output ³

Note 1: Only arcs that are circular (where primary axis and secondary axis are equal length) are processed by MRF Intersect.

Note 2: Including non-circular arcs.

Note 3: These elements are not processed.

2.6 MRF Duplicate Remover

MRF Duplicate Remover is used to remove duplicate elements from a two-dimensional design file. There are two options for identifying duplicates:

Option 1 (default, By-Criteria): elements are considered to be duplicates if they meet user-specified criteria such as identical level, color, weight, style, graphic group number, property bits, attribute (database) linkages, zhigh, zlow, file position and geometry.

Option 2 (By-geometry): the elements are considered to be duplicates if their geometries are within tolerance.

Text and text node elements must have identical text strings to be considered duplicates. If the design file does contain duplicate elements, only the elements with a smaller tolerance will remain after running MRF Duplicate Remover. If two or more elements have the same tolerance, the element with the smallest file offset will be retained and the other(s) will be deleted. The deleted duplicate elements will be written to a separate deleted elements file.

2.6.1 Elements Processed by MRF Duplicate Remover

MSTN Element Type	MSTN Element Number	Duplicate Removal Processing
CELL_HEADER_ELM ²	2	Candidates for removal
LINE_ELM ²	3	Candidates for removal
LINE_STRING_ELM ²	4	Candidates for removal
SHAPE_ELM	6	Candidates for removal
TEXT_NODE_ELM	7	Candidates for removal
CURVE_ELM	11	Candidates for removal
CMPLEX_STRING_ELM	12	Candidates for removal
CMPLEX_SHAPE_ELM	14	Candidates for removal
ELLIPSE_ELM	15	Candidates for removal
ARC_ELM ²	16	Candidates for removal
TEXT_ELM	17	Candidates for removal
SHARED_CELL_ELM	35	Candidates for removal
ALL OTHER ELEMENTS		Copied to output ¹

Note 1: If no output file is specified, these elements will be left in the input file.

Note 2: In the combined module of MRF Clean, lines, line strings, arcs and cells are the only elements processed for duplicate removal.

2.7 MRF Join

In many mapping and GIS applications, there is a need to join singly-connected elements into longer ones. The benefits of joining include:

1. Reduced design file size

2. The number of geographic features is reduced, making many mapping/GIS operations more efficient - there will be fewer elements to edit or attach database attributes to.

MRF Join can correctly process only clean 2D design files. If a design file is not clean to begin with, first run the other MRF Clean modules necessary to make it so (see *Workflow Recommendations* below).

A pair of linear elements become candidates for joining when only the two of them are singly connected at a given node or end point. MRF Join will join them to form one longer element if they match in terms of the specified criteria, which might include having the same:

- level
- symbology (color, line weight and line style)
- graphic group number
- property bits
- attribute linkages
- zhigh
- zlow



Note: When checking database linkages, MRF Join will ignore remote and modified attribute linkage property bits.

2.7.1 Elements Processed by MRF Join

MSTN Element Type	MSTN Element Number	MRF Join Processing
CELL_HEADER_ELM	2	Joining can be stopped at Cells
LINE_ELM	3	Candidates for joining
LINE_STRING_ELM	4	Candidates for joining
ARC_ELM ¹	16	Candidates for joining
ALL OTHER ELEMENTS ²		Copied to output ³

Note 1: Only arcs that are circular (where primary axis and secondary axis are of equal length) are processed by MRF Join. In the MRF Clean combined module, arcs will not be joined because complex chains will not be created in combined MRF Clean.

Note 2: Includes non-circular arcs.

Note 3: If no output file is specified, these elements will be left in the input file.

2.8 MRF Conflate

Conflation changes the geometry of one linear element to match that of another, if the two are approximately the same to begin with.

2.8.1 Elements Processed by MRF Conflate

MSTN Element Type	MSTN Element Number	MRF Conflate Processing
LINE_ELM	3	Candidates for conflating
LINE_STRING_ELM	4	Candidates for conflating
ARC_ELM ¹	16	Candidates for conflating
ALL OTHER ELEMENTS ²		Copied to output ³

Note 1: Only arcs that circular (where primary axis and secondary axis are of equal length) are processed by MRF Conflate.

Note 2: Includes non-circular arcs.

Note 3: If no output file is specified, these elements will be left in the input file.

2.9 MRF Dangle Processor

The elements processed by MRF Dangle Processor include line, line strings and arc elements. Cells are used merely to determine whether an element is a dangle or not. Other elements are copied directly to the output file.

Dangle elements (dead-ends) will be either flagged with a letter “D” at the user specified level or deleted (if the delete option is chosen).

2.9.1 Elements Processed by MRF Dangle Processor

MSTN Element Type	MSTN Element Number	MRF Dangle Processing
CELL_HEADER_ELM	2	Candidates for processing ⁴
LINE_ELM	3	Candidates for processing
LINE_STRING_ELM	4	Candidates for processing
ARC_ELM ¹	16	Candidates for processing
ALL OTHER ELEMENTS ²		Copied to output ³

Note 1: Only arcs that are circular (where primary axis and secondary axis are of equal length) are processed by MRF Dangle Processor.

Note 2: Includes non-circular arcs.

Note 3: If no output file is specified, these elements will be left in the input file.

Note 4: Lines, line strings and arcs may **stop** at cells.

3. Running MRF Clean

3.1 Installation

To install MRF Clean for MicroStation V8, simply run the MMTK setup program and follow the instructions (please remove all previous MMTK installation before running the setup program) or copy the following files into the corresponding MicroStation folders:

mrfclean.ma mrfflag.ma clean8.dll mrf_license.dll getHostID.exe	Copy to the “mdlapps” folder
Examples (folder)	Copy to any folder. They are just example files
MrfClean.hlp MrfClean.cnt MMTK_84 for V8.pdf	Copy to “docs” folder

You need to contact MRF representatives to obtain a valid license before you can run the program. To run MRF Clean in MicroStation V8, simply key in:

“Mdl Load Mrfclean”

3.2 Main Dialogue Box

As with previous MRF Clean versions, a number of input parameters are required to successfully clean vector data. This section discusses how each parameter affects processing.

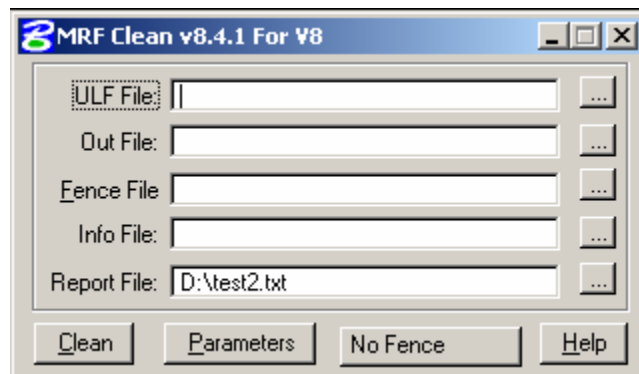


Figure 3.2.1 Main Dialog Box of MRF Clean

ULF File: (Optional)

This is used to select a Universal List File for processing. If a ULF file is specified, the corresponding default design file will be opened and the elements in it processed. The output will be placed in the output file specified or written back to the original ULF design file, if no output file has been specified.

Out File: (Optional)

By default, the active design file is the output file; if no output file is specified, the result of the cleaning process will be written back to the design file. If an output file is specified, the results (both processed and unprocessed elements) will be written to that file, instead.



Recommendation: It's good practice to always specify an output file - and keep the original design file intact. If you specify a file that already exists, an alert dialog box will pop up and ask for permission to overwrite. In batch mode, MRF Clean will not overwrite an existing file.

Fence File: (Optional)

Specify a fence file name if you have selected “**Use Fence File**” (see section for “**Fence Option**” below). The *shape element* found in the file will be used as a fence for the process.

INFO File Key Words: FENCE_FILE

Info File: (Optional)

An ASCII-text parameter file, as shown below, can be used to automatically set some or all of the process parameters.



Note: Check-box values are represented as “0” for OFF or “1” for ON in an info file.



Note: Option buttons can have as many values as the number of options: the first option is “1”, the

second is “2”, and so on. The ordering of the options is the same as displayed in the dialog box.

The INFO File has the following format.

```
# Copyright: MRF Geosystems
# INFO file for MRF Clean.
# Specifies all the required parameters:
FENCE_OPTION:1
FENCE_FILE:
CURVE_FACTOR:2.000000
DANGLE_FACTOR:1.000000
FILTER_FACTOR:1.000000
FREE_END_FLAG_LEVEL:63
INTERSECTION_FLAG_LEVEL:
SUB_TOL_ELE_FLAG_LEVEL:
FLAG_FONT_NUMBER:3
SAVE_INFO_FILE:
REMOVE_DUPLICATES_BY_CRITERIA_OR_GEOMETRY:1
USE_CELL_AS_NODE_OR_NON_NODE:1
BY_LEVEL:0
STROKE_CIRCULAR_ARCS:0
FUZZY_INTERSECTION:1
TRUE_INTERSECTION:1
DEL_SUB_TOL_ELE:1
DO_STROKE:0
DO_SIMPLIFY:0
DO_EXTENSION:0
DO_SHORT_ELE:0
DO_INTERSECTION:1
DO_REMOVE_DUP:1
DO_JOIN:1
DO_CONFLATION:0
DO_REMOVE_DANGLE:1
CRITERIA_LEVEL:1
CRITERIA_COLOR:1
CRITERIA_STYLE:1
CRITERIA_WEIGHT:1
CRITERIA_GRAPHIC_GROUP:1
CRITERIA_PROPERTY_BITS:1
CRITERIA_LINKAGES:1
CRITERIA_ZHIGH:1
CRITERIA_ZLOW:1
CRITERIA_FILEPOS:1

[LEVELS]

LEVEL_1:0.03
LEVEL_2:0.05
```

You can generate an info file through a text editor or by pressing the “Save Info File” button in the **MRF Clean Parameter** dialog box.

While the format of the INFO file is important, the ordering of the parameters is not. Parameter keywords must begin at the left margin and be followed by a full colon (‘:’). Whatever follows the colon is the value assigned. There should be nothing else on the line. Any line beginning with a ‘#’ is treated as a comment and disregarded.

Note that keywords **are** case-sensitive.



Note: Once an INFO file is specified, MRF Clean will load the parameters from the INFO file and overwrite existing parameters.

Report File:

By default, an ASCII text report file having the design file name but with the extension '.txt' will be created in the default directory. The report will contain information such as parameter settings, processing time, elements processed, and errors that occurred. If the target file already exists, the report will simply be appended to it.



Note: Use the File Picker buttons beside the “ULF File”, “Out File”, “Info File”, and “Report File” fields to call up a dialog box for specifying existing files or directory locations.

Parameters:

When you click this button, the “**MRF Clean Parameters**” sub-dialog box (see *Figure 3.2.2*) appears. Parameter values are read in from the INFO file (if designated) or set to defaults.

Fence Option: (option button) (Default is “No Fence”)

There are three settings available:

1. No Fence (specify “1” in INFO file)
Process the entire design file. In this mode, an advanced automated partitioning algorithm is employed to generate a set of internal fences to optimize processing efficiency for large files.

Note: Internal fences are NOT generated if the “**By Level**” option is enabled.

2. Use Current (specify “2” in INFO file)
Process the elements inside or overlapping the current fence. Use this when only a small area needs cleaning (use the MicroStation “place fence block” or “place fence shape” tool to define the fence). This option is not available in batch mode.
3. Use Fence File (specify “3” in INFO file)

This is similar to option 2, but the fence used is actually a shape in a design file. When this option is selected, a file selector will pop up allowing you to select an existing design file containing a single shape. The file specified should have the same settings as the file to be cleaned¹. If you press “Cancel” in the file selector dialog, the fence option will be re-set to “**No Fence**”.

INFO File Key Words: FENCE_OPTION

Help:

The help button invokes the help file utility with information on input parameters and operating instructions.

Clean:

After all tolerances, modules, options and factors have been set, this button is clicked to begin the cleaning process.

3.2.1 How to Specify Input Parameters

Input parameters can be specified manually, or through the use of an INFO file. Click on the “**Parameters**” button on the main dialog to bring up the dialog for manual parameter entry.

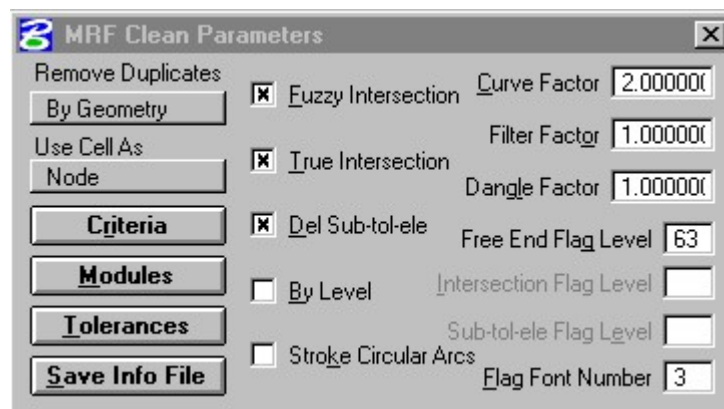


Figure 3.2.2. Combined Module Parameters Dialog Box

¹ The easiest way to do this is to use the file to be cleaned as the “seed” for the fence file.

(a) Tolerances Button:

This button opens the “*Setup Tolerances*” dialog box (Figure 3.2.3). The default tolerance value is -0.01 . Tolerances should be specified in **Master Units (MU)**. Only levels with non-negative tolerances will be processed. The “*Setup Tolerances*” dialog box shows all levels containing valid elements along with each level’s current tolerance setting.

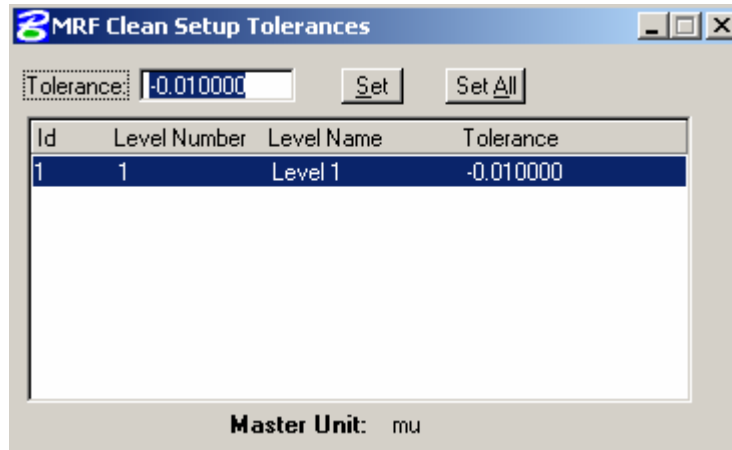


Figure 3.2.3 *Setup Tolerances*

There are two buttons for setting parameters: “**Set**” and “**Set All**”. “**Set**” is used to set the tolerance for the selected level, while “**Set All**” sets the tolerances for all of the levels at once.

To change the tolerance values, type the required tolerance value in the “**Tolerance**” field and click on one of the buttons. The list box will automatically show the new settings.



Note: Once the “*Setup Tolerances*” sub-dialog box is opened, all changes made will be reflected in subsequent processing.

INFO File Key Words: [LEVELS]
 LEVEL_1:
 LEVEL_2:
 etc...

If specified in an info file, the tolerances for the various levels should be preceded by the keyword ‘[LEVELS]’ (on a line by itself). For example, setting a tolerance of 0.05 for level 1 and a tolerance of 0.11 for level 3, it would require the following lines in the parameter file:

[LEVELS]
LEVEL_1: 0.05
LEVEL_3: 0.11

(b) Criteria Button:

This button opens a dialog box (*Figure 3.2.4*) used to set criteria for duplicate removal, and conflation.



Figure 3.2.4 Conflate, and Duplicate Remove Criteria

- (i) Duplicate Removal: MRF Clean provides the option to remove duplicate elements that have the same geometry and meet the criteria that the user has specified.
- (ii) Conflation: The conflation process within MRF Clean may create duplicate elements. Such elements will be processed in the same manner as in the “Duplicate Removal” process.

The criteria that the user can specify are:

- Level
- Color
- Style
- Weight
- Graphic Group
- Property Bits
- Attribute Linkages (all)
- Zhigh
- Zlow
- File Position

By default, all element attribute criteria are selected. Except for 'File Position', if an attribute is selected, two elements cannot be considered duplicates unless they have the same value for that attribute.



Note: File position is used only in duplicate removal. If specified, the element with the smallest file position is kept (i.e., the element first placed in the file) when there are duplicate elements.

INFO File Key Words: CRITERIA_LEVEL:
 CRITERIA_COLOR:
 CRITERIA_STYLE:
 CRITERIA_WEIGHT:
 CRITERIA_GRAPHIC_GROUP:
 CRITERIA_PROPERTY_BITS:
 CRITERIA_LINKAGES:
 CRITERIA_ZHIGH:
 CRITERIA_ZLOW:
 CRITERIA_FILEPOS:

(c) Modules Button:

This button opens a “**Module Selection**” dialog box (*Figure 3.2.5*) for choosing the modules to run on the current design file.

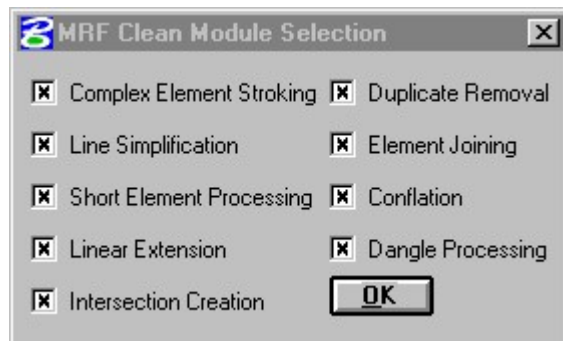


Figure 3.2.5. Modules dialog box

INFO File Key Words: DO_STROKE:
 DO_SIMPLIFY:
 DO_SHORT_ELE:
 DO_EXTENSION:
 DO_INTERSECTION:
 DO_REMOVE_DUP:
 DO_JOIN:
 DO_CONFLATION:
 DO_REMOVE_DANGLE:

(d) Save Info File Button:

Saves the current parameter settings to the specified file.

INFO File Key Words: SAVE_INFO_FILE:

(e) Curve Factor:

Non-linear elements (circular arcs optionally) are approximated by vectors with a stroking distance of ($CURVE_FACTOR * tolerance$). The stroking distance is the maximum distance between the curve and the approximating vectors. The $CURVE_FACTOR$ field has a default value of 2.0, which means that the selected curves will be stroked into line strings using ($2.0 * tolerance$) as the stroking distance. You can specify a value for the $CURVE_FACTOR$; its minimum value is 0.01.

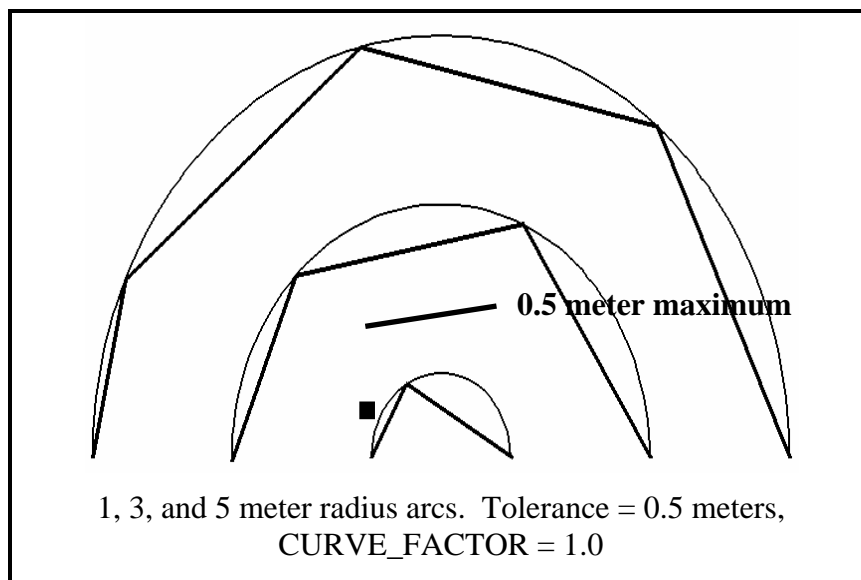


Figure 3.2.6. Arc stroking results

The degree of smoothness of the stroking is a function of the $CURVE_FACTOR$, the tolerance, and the radius of the curve (i.e., *Figure 3.2.6*). Smaller curve factors will always produce smoother stroked linestrings, but smaller curve factors will also produce larger elements with more vertices. You might need to experiment with your own data to develop the best set of parameters.

INFO File Key Words: $CURVE_FACTOR$:

(f) Dangle Factor: (Default is 1.0)

A dangle is an element that has at least one free end point. MRF Clean removes a dangle if its length is less than ($Dangle\ Factor * tolerance$). The minimum value for Dangle Factor is 0.0. This principally affects removal of overshoots. Some

dangles, such as street centerlines ending at a cul-de-sac, may be appropriate to keep.

INFO File Key Words: DANGLE_FACTOR:

(g) Filter Factor: (Default is 1.0)

MRF Clean allows you to weed the input elements, (i.e. to simplify line-work by removing unnecessary vertices). MRF Clean uses a modified version of the Douglas-Peucker algorithm for this process. The modification produces better results for some special cases. This process uses a weeding tolerance of (Filter Factor * tolerance). The minimum value for Filter Factor is 0.0. The larger the value of (Filter Factor * tolerance), the more vertices will be removed. Here again, you might need to experiment with your own data to develop the best set of parameters.

INFO File Key Words: FILTER_FACTOR:

(h) Free End Flag Level: (Optional. Default Level is 63)

If you specify a valid level in this field, MRF Clean will place a text element “D” on that level to mark the location of each free end not resolved as an overshoot or undershoot. The text is created using current MicroStation settings for text size, font, weight, and justification. MRF Flag Editor, another component of the MRF Mapping Tool Kit, can be used to efficiently edit the marked free ends.

INFO File Key Words: FREE_END_FLAG_LEVEL:

(i) Intersection Flag Level: (Optional. Default Level is 62)

If set to a valid level, the locations where true intersections would normally be created are indicated with the text element “X” (on the level specified). The text is created using current MicroStation settings for text size, font, weight, and justification. MRF Flag Editor can then be used to locate the marked intersections. “True Intersections” must be off for this parameter to have effect.

INFO File Key Words: INTERSECTION_FLAG_LEVEL:

(j) Sub-tol-ele Flag Level: (Optional. Default is level 61)

If set to a valid level, the start points of sub-tolerance elements are indicated with the text element “S” (on the level specified). The text is created using current MicroStation settings for text size, font, weight, and justification. MRF Flag Editor can then be used to locate the marked elements. “Del_Sub_tol_ele” must be off for this parameter to have effect.

INFO File Key Words: SUB_TOL_ELE_FLAG_LEVEL:

(k) Flag Font Number: (Default is 3 [engineering])

Specifies the font to be used for flag characters. This parameter can also be specified in the INFO file. You can always check for the fonts available within the system by clicking the “Text” item under “Element” tool in MicroStation.

INFO File Key Words: FLAG_FONT_NUMBER:

(l) Remove Duplicates: (check-box) (Default is “By Geometry”)

Exact duplicate elements may be created as a result of splitting. Two options are available for removing these duplicates:

1. By Criteria (specify “1” in INFO file)

Duplicate elements that have the same geometry and the same values for the selected MicroStation element characteristics (refer to *MRF Clean criteria*) are removed .

2. By Geometry (specify “2” in INFO file)

Duplicate elements are removed if they have the same geometry, regardless of their attributes (criteria). The duplicate element with the smallest tolerance will be kept. If a number of elements have the same tolerance: only one of them will be kept.

INFO File Key Words:

REMOVE_DUPLICATES_BY_CRITERIA_OR_GEOMETRY:

(m) Use Cell As: (check-box) (Default is Node)

There are two settings: node and non-node. Even though a cell is a point object, it may also be treated as a node, or line end, for various purposes, such as dangle removal, element joining, conflation, and free end flagging.



Note: This function works on both point and graphic types of MicroStation cells.

1. Non Node (specify “1” in INFO file)

Cells are ignored in element joining, dangle removal, conflation, and free end flagging.

2. Node (specify “2” in INFO file)

Cells are treated as nodes in element joining, dangle removal, conflation, and free end flagging. In some situations, a line ending at a cell is not considered be dangle. In such cases, the cell should be treated as a node.

For example, in *Figure 3.2.7*, if the cell at point C is treated as a node, then line AC will not be considered as a dangle. However, if the cell at point C is not treated as a node, then line AC will be treated as a dangle.

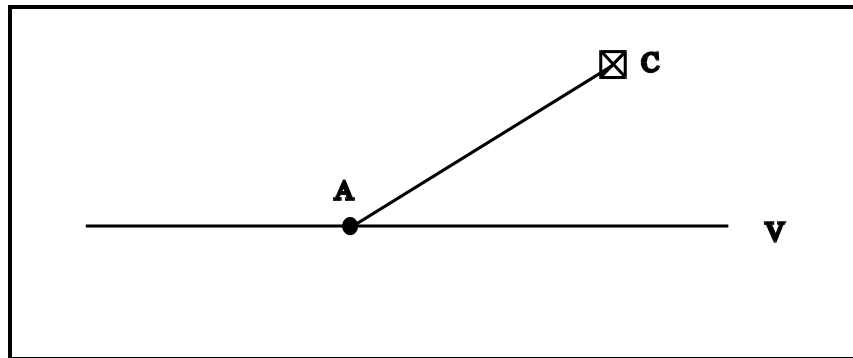


Figure 3.2.7. Cell Usage Example

INFO File Key Words: USE_CELL_AS_NODE_OR_NON_NODE:

(n) By Level: (check-box) [default is disabled]

If enabled, level with a valid tolerance will be processed separately. Otherwise, those levels will be processed together.

INFO File Key Words: BY_LEVEL:

Note: Internal fences are NOT generated when the By Level option is enabled.

(o) Stroke Circular Arcs: (check-box) [default is disabled]

If enabled, circular arcs will be stroked to line string elements before processing.

INFO File Key Words: STROKE_CIRCULAR_ARCS:

(p) Fuzzy Intersection: (check-box) [default is enabled]

When enabled, fuzzy intersections can be created (i. e., over-shoots and under-shoots will be corrected when within tolerance). MRF Clean will create intersections where line-work is very close to intersecting, but not actually touching. The fuzzy intersections are useful in correcting undershoots and overshoots.

INFO File Key Words: FUZZY_INTERSECTION:

(q) **True Intersection:** (check-box) [default is enabled]

When enabled, true intersections can be created (i.e. line-work will be broken at the point where the line-work crosses another element or itself, and separate linear elements are created). Enabling this option disables intersection flagging (see “**Intersection Flag Level**” above).

INFO File Key Words: TRUE_INTERSECTION:



Note: MRF Intersect does *not* flag fuzzy intersections.



TIP: The best set of parameters to use for flagging, and not breaking line-work at true intersections is:

Toggle True Intersection to OFF

Toggle Fuzzy Intersection to OFF

Set “Intersection Flag Level” to a valid level

(r) **Del Sub-tol-ele:** (check-box) [default is disabled]

When enabled, elements that are shorter than the tolerance for the level they appear on will be deleted. Enabling this option disables sub-tolerance element flagging (see “**Sub-tol-ele Flag Level**” above).

INFO File Key Words: DEL_SUB_TOL_ELE:

3.3 Running MRF Clean - Step-by-Step

After beginning a MicroStation session, follow these steps to load MRF Clean:

1. Start MicroStation and open the design file to be processed.
2. Choose *MDL Applications* from the *Utilities* pull-down menu.
3. Select mrfclean.ma from the list of available applications.
4. Click the Load button.

 **TIP:** Instead of using Steps 2 to 4 to load MRF Clean, you can key in the following on the MicroStation command line:

mdl load mrfclean.ma

Optional parameters of Mrfclean key-in

-i[infofile]: this option automatically loads the specified info file. Ex. Mdl Load Mrfclean -ic:\test.inf

-r[reportfile]: this option specifies the report file when Mrfclean loads. Ex. Mdl load mrfclean -rc:\report.txt

-o[outputfile]: this option specifies the output file when Mrfclean loads. Ex. Mdl load mrfclean -oc:\output.dgn

With the software loaded, the following steps are taken to begin file processing:

5. Specify the full path for the output design file in the “**Out File**” text field. Alternatively you can use the button to the right of the text field to call up the file dialog and create a file in an existing directory. If the file specified already exists, a warning will be given and MRF Clean will request permission to overwrite the existing file. If no output file is specified, MRF Clean will overwrite the existing input file.
6. Specify an existing **ULF** File if required.
7. If desired, enter the name of a valid parameter file (including path if necessary) in the “**Info File**” field. The information in the file will be used to set some or all of the process parameters.
8. Enter a report file name in the “**Report File**” field with the full path. If no report file is specified, a report file with the same name as the input design file, but with the extension ‘.txt’ will be created in the current directory.
9. Select the appropriate “**Fence Option**” (no fence, use current fence, or use fence file).
10. Press the “**Parameters**” button to open the parameters dialog box and set all appropriate options/parameters for the cleaning process.
11. Press the Tolerances button and, in the dialog that appears, enter and/or modify the tolerance for each level requiring processing. To set the same tolerance for all levels, use the “**Set All**” option. To set individual tolerances, use the “**Set**” button.



Note: Tolerances are specified in the Master Units of the design file being processed. Only levels assigned a non-negative tolerance value will be processed.

12. Press the **“Modules”** button and, in the dialog that appears, select the MRF Clean modules you want to run. Each module can be toggled on/off by clicking on the check box to the left of the module name.

If Duplicate Removal, Conflation, and/or Element Joining modules are selected in step 12, Press the **“Criteria”** button to select the parameters to be considered in identifying candidate elements for Duplicate Removal, Conflation, and/or Element Joining.

13. Close the parameters dialog box (press the ‘x’ in the upper right-hand corner of the box) and press the **“Clean”** button on the main dialog to begin the cleaning process.

If any of the input parameters are invalid, MRF Clean will stop processing and display an error message in the **“Error”** field of the MicroStation command window.

4. Batch process

Using batch command processing can provide additional efficiency for processing a large number of files without requiring constant attention, or for increasing equipment usage by processing on evenings and weekends. Batch commands can also be used to automate recurrent tasks.

To run the batch process, simply key in the following in the Microstation command line:

```
Mdl load mrfclean -b[batch parameter file]
```

```
Ex. Mdl load mrfclean -bc:\batchprocess.txt
```

The batch parameter file should contain lines of text in the following format:

```
<input dgn><output dgn><info file><report file>
```

Ex.

```
<c:\input1.dgn><c:\output1.dgn><c:\input1.inf><c:\report1.txt>
```

```
<c:\input2.dgn><c:\output2.dgn><c:\input2.inf><c:\report2.txt>
```

```
<c:\input3.dgn><c:\output3.dgn><c:\input3.inf><c:\report3.txt>
```

You can also load Microstation and run the batch process automatically through DOS prompt or a Windows batch job with the following command line:

```
[Microstation command] [input dgn] -s[script file]
```

```
Ex. "C:\program files\Bentley\Program\MicroStation\ustation.exe" c:\input1.dgn -  
sc:\cleanactions.txt
```

In the script file (c:\cleanactions.txt), put the MRFClean batch command.

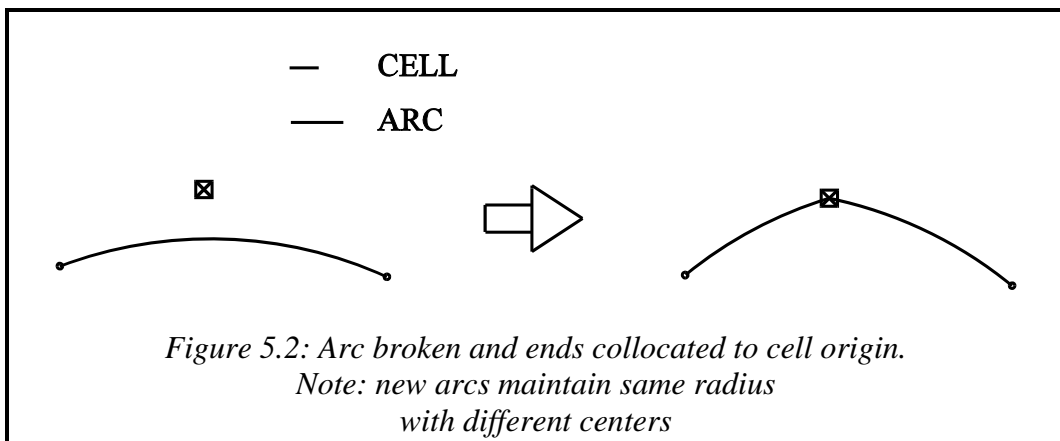
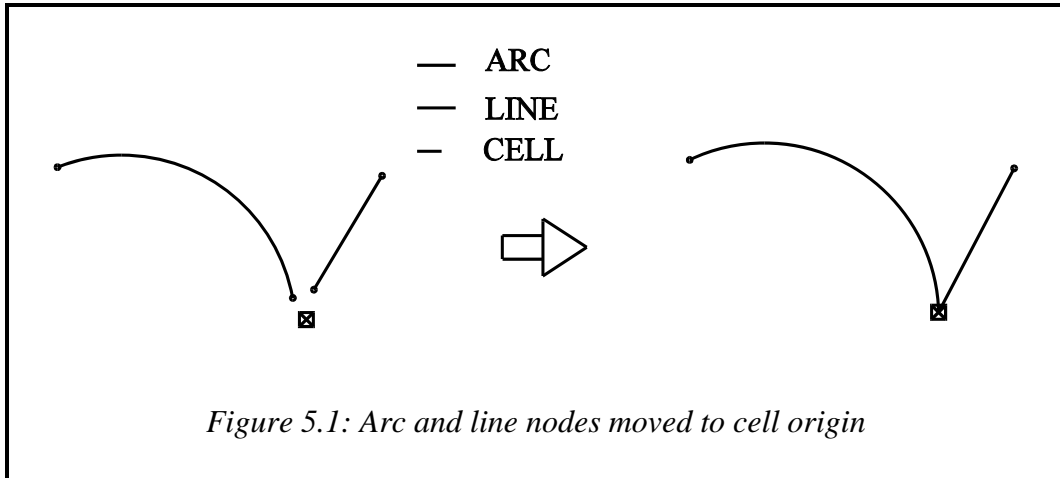
```
Ex. Mdl load mrfclean -bc:\batchprocess.txt
```

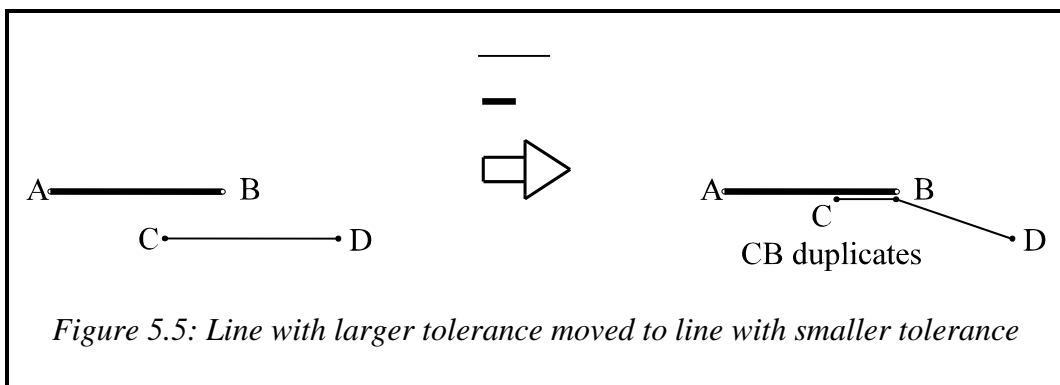
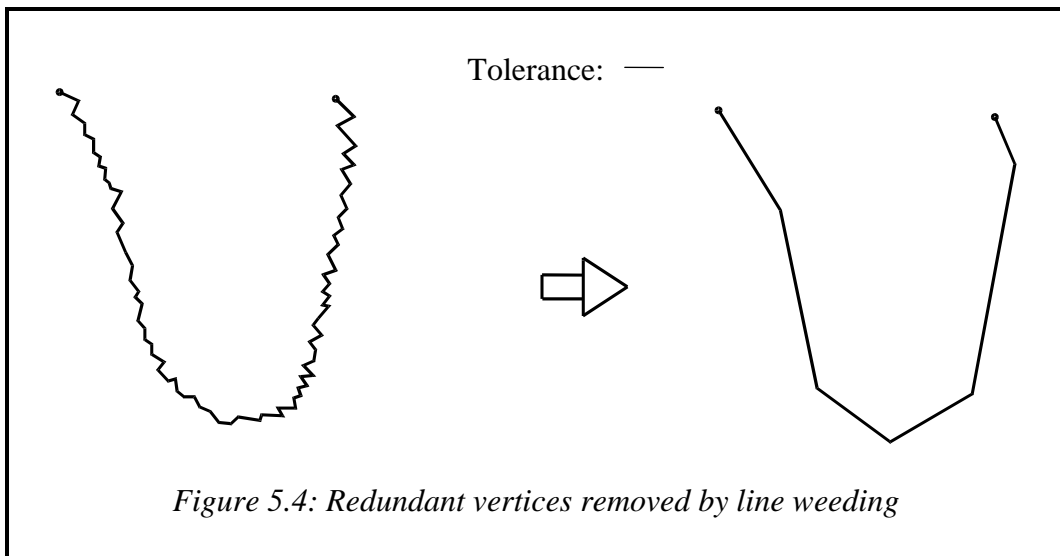
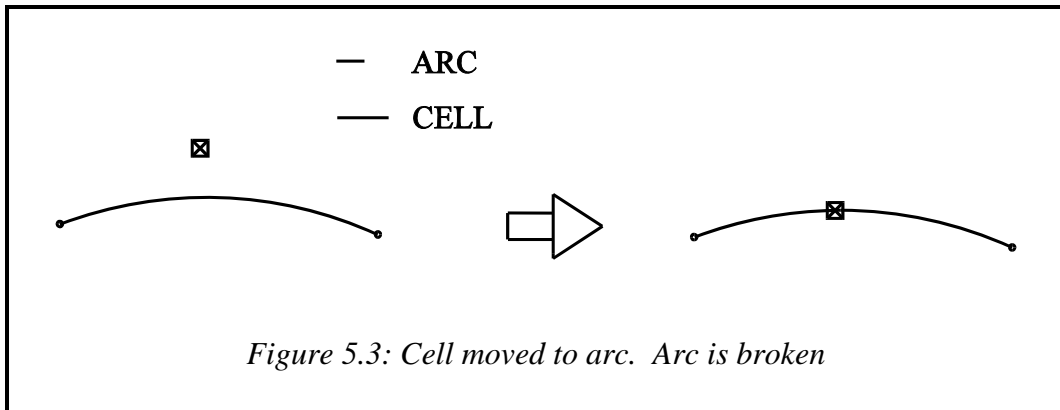
The batch process will run as soon as Microstation loads.

5. Sample Results of MRF Clean

MRF Clean uses generic algorithms to perform data cleaning and does not follow a set of pre-defined cases. The best way to discover the behavior of MRF Clean is to construct test cases and run MRF Clean on them. The following diagrams will help

you understand the cleaning principles of MRF Clean. Each element is assumed to be on a unique level unless otherwise stated. Tolerances are shown at the top of each figure.





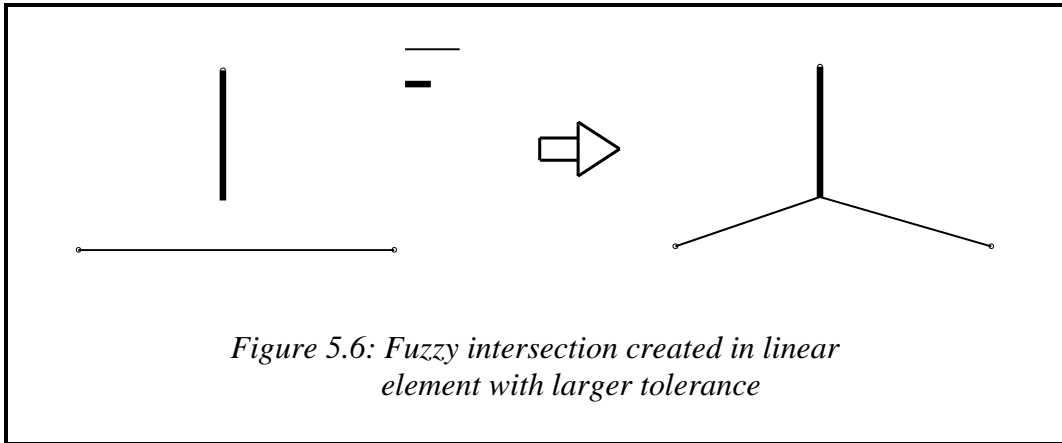


Figure 5.6: Fuzzy intersection created in linear element with larger tolerance

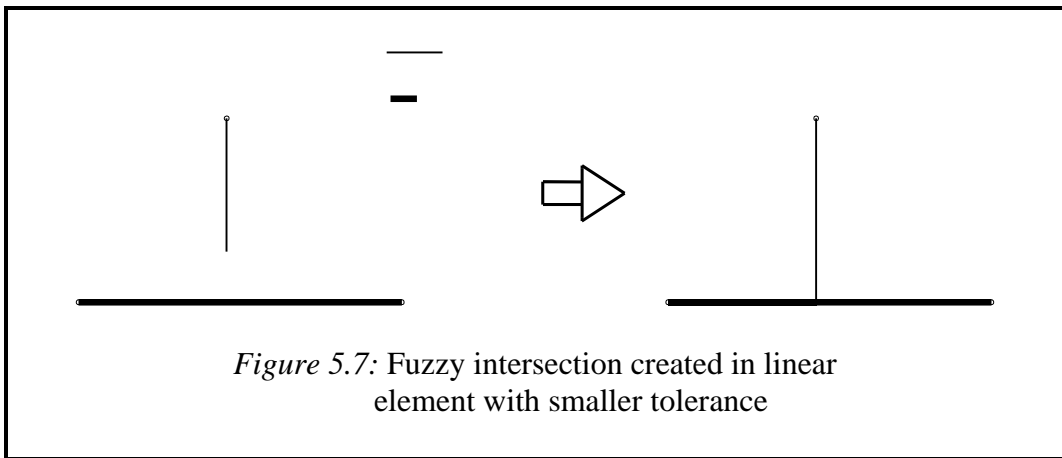


Figure 5.7: Fuzzy intersection created in linear element with smaller tolerance

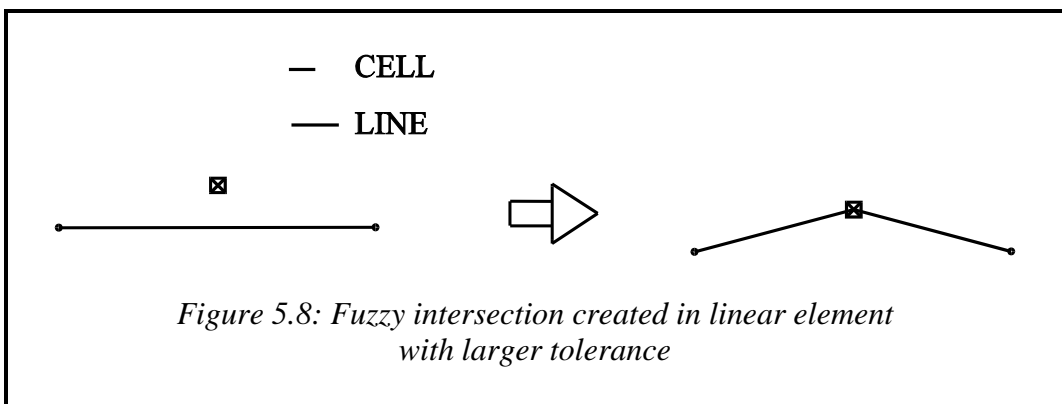
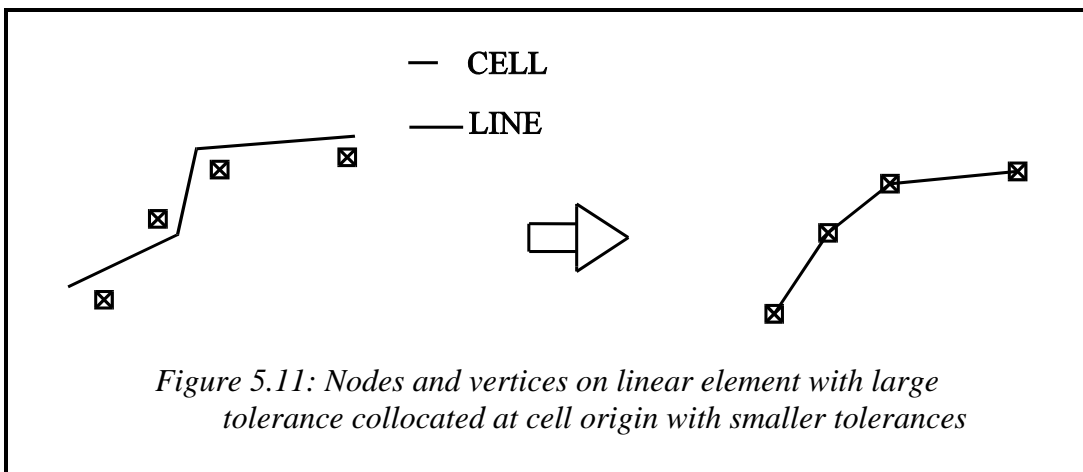
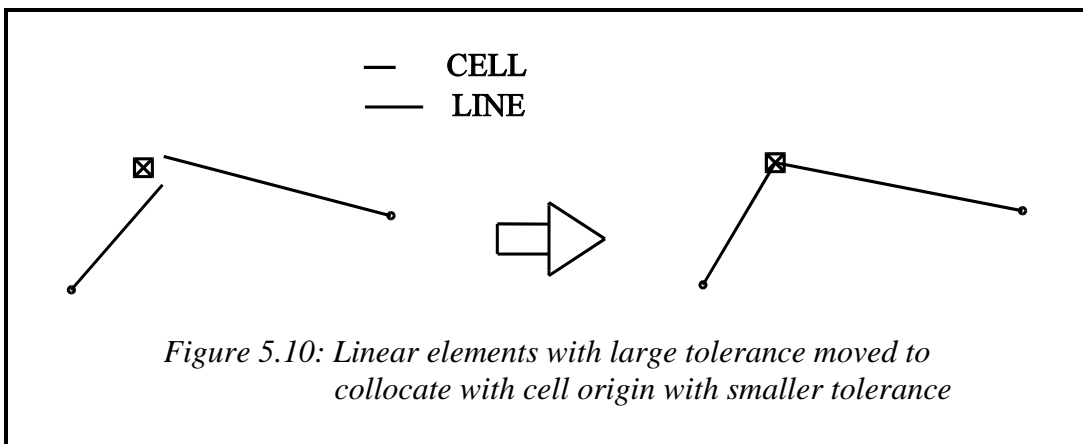
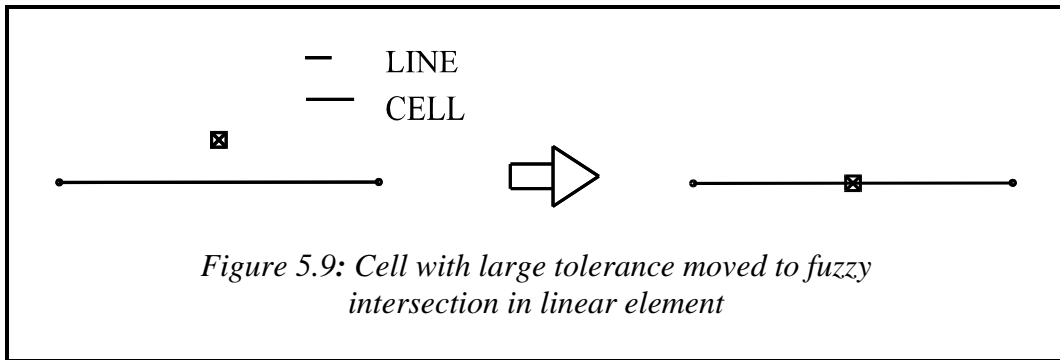


Figure 5.8: Fuzzy intersection created in linear element with larger tolerance



6. Workflow Recommendations

MRF Clean can be operated as a single product through the combined module and integrated user-interface. This section discusses the workflow used in the combined module, and processing tips for maximum efficiency.

6.1 General Processing Tips

There are several fundamental tips and tricks for optimizing MRF Clean processing. The following list identifies and briefly describes some key issues to consider when cleaning map data.

Know Your Data

The first step in any cleaning exercise is to become familiar with your source data. Information on data quality (i.e., 1m vs. 100m accuracy), data currency, and intended use is important in determining which cleaning modules and tolerances should be used. If such information is not available, a visual inspection of the design file(s) should provide insight into average line-work gap sizes, line weeding requirements, and other issues which may exist.

Get Into the Habit

It's a good idea to always use an output file, and not overwrite the input design file. The primary reason for this data protection. In the event that a tolerance or criteria setting is incorrect, the input data is maintained and can be re-processed with the correct parameters.

Start Small

When setting cleaning tolerances, it is always best to start small. With smaller tolerances, the software uses a smaller search radius, which reduces the number of potential element intersections to consider and increases processing speed. Also, if the bulk of the line-work errors can be corrected using a small tolerance, more detail can be maintained in the data set. One or more cleaning processes can always be repeated with larger tolerances to increase the number of errors automatically corrected.

Mix It Up

Depending on the source data set, and its intended use, you may achieve better results running the individual modules with different tolerances. This capability is one of the reasons for the modular nature of this latest version of MRF Clean.

Batch the Process

Batch processing can greatly enhance productivity by taking advantage of machine down time (i.e., evenings/weekends). Parameter files and report files can be used as

efficient data quality assurance tools to monitor the input and results of batch operations. File management tools (e.g., use and deletion of temporary files) can be built into the batch process. Also, if MRF Clean is used with other MRF software, map cleaning, polygon formation, buffer zone generation, and NAD conversion can be combined for complete custom workflows.

6.2 Workflow Used in the Combined Module

The combined module is designed to ease the transition for users of earlier versions to this latest release of MRF Clean. The MRF Clean v8.x workflow is illustrated in *Figure 6.3.1*. In version 8, all cleaning functions were performed on each file processed - there was no way of skipping over an unnecessary step.

By default, the combined module workflow will be the same as that in version 8 of MRF Clean. But, now individual processes may be by-passed as suits the particular project or data set. The extension module, if enabled, will be run after line simplification and before intersection.

6.3 Custom Workflow Creation

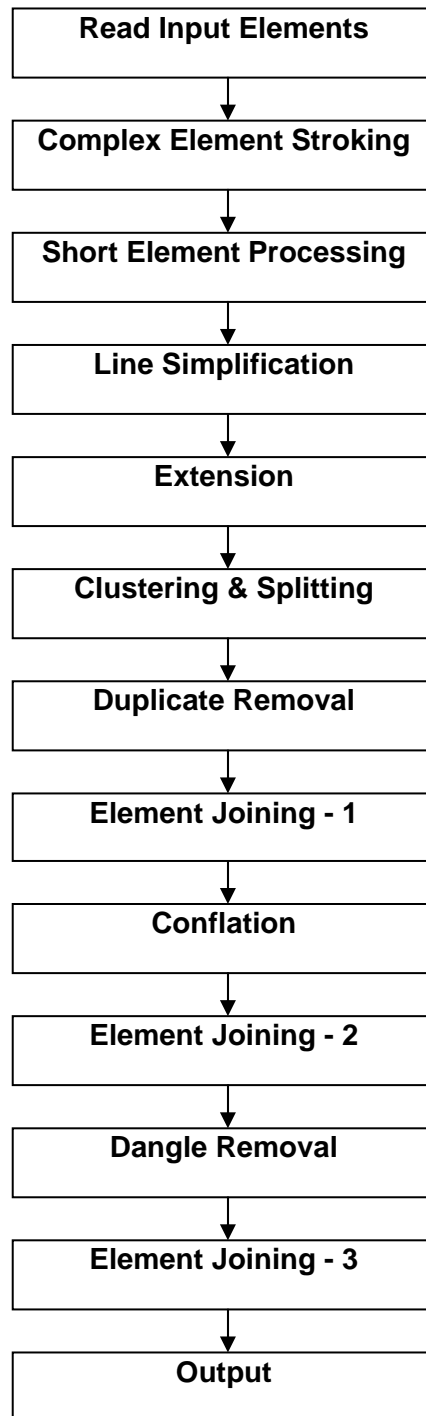


Figure 6.3.1 MRF Clean combined module workflow

Using the nine individual modules of MRF Clean it is possible to create any number of customized workflows for specific projects and/or data sets. It is important, however, to understand the data being processed and the desired end result.

To illustrate two possible approaches to this, two example data sets are presented. The first is an example of municipal cadastral data while the second involves resource mapping for forestry applications. The use each application makes of the various modules is summarized in tabular form.

6.3.1 *Example 1 – Cadastral Information*

In cadastral applications, data quality is a primary concern. The following paragraphs describe workflow considerations for cadastral data and how the modules within MRF Clean can be used to best correct and maintain this information.

Data collected for cadastral mapping is often collected using range and bearing information. For linear property boundary, the bearing is a critical parameter. The *extension* module allows for the maintenance of bearing while correcting gaps in digitized or COGOed line work. Also, because surveyed data does not generally contain line work defined by too many points, *line weeding* may not be desirable. Following the linear extension pre-processing, *intersection creation* - a critical step in any cleaning operation - is performed to correct overshoots and undershoots. Another cleaning module which is often not desired in cadastral applications is *element joining*. Many applications require that a lot or block be defined by four or more individual line segments, rather than one or more joined line strings. Also, arcs contain valuable information which may be lost by creating a complex element in the joining process. *Duplicate removal* can be an important quality assurance process step for any data set, and MRF Clean's criteria settings can be leveraged to protect duplicate geometry, which must be maintained on separate levels. *Conflation* can be used with cadastral data to align lot/block/plan boundaries which share end nodes but have errors in their defining vertices. Finally, the *dangle processor* is helpful in eliminating dangling elements shorter than the specified tolerance(s) and flagging remaining errors for queued-editing with MRF Flag Editor. See Table 6.1 below for a summary of cadastral recommendations.

Table 6.1. Modular summary for cadastral data processing.

MRF Clean Module	Recommended for Use?	Comments
Complex Element Stroking	No	To preserve embedded arcs, drop complex before cleaning.
Line Simplification	No	Unless line work is generated from scanned or poorly digitized sources.
Short Element Processing	Yes	Beware of MicroStation points defined as Zero-length lines.
Extension	Yes	Key to maintaining bearing information.
Intersection Creation	Yes	Ensures connectivity and closure.
Duplicate Removal	Yes	Use criteria to refine element comparisons (i.e., same geometry, and different levels).
Element Joining	No	Often not desirable for cadastral data.
Conflation	Yes	Use criteria to refine process.
Dangle Processing	Yes	Flags remaining errors for queued-editing with Flag Editor.

6.3.2 Example 2 – Resource Mapping

Resource mapping applications must deal with the issues of digitizing errors, polygon closure, and design file inefficiency. Resource data is commonly acquired over large areas from aerial photography or hard-copy data capture. The following paragraphs describe workflow considerations for resource mapping applications, and how the modules within MRF Clean can be used to best correct and maintain this information.

Data compiled for resource mapping often contains information digitized from a variety of sources. Because of this, there are a number of issues of data quality and operating efficiency to be considered. Pre-intersection processes such as complex element stroking, line weeding, and short element deletion can be used to reduce design file size and speed later processing. *Linear extension* is not recommended for resource data as errors in line-work are generally random in direction and can be best handled by the *intersection creation process*. *Intersection creation* - a critical step in polygon closure and network connectivity (e.g., roads, streams) - is performed to correct overshoots and undershoots, and break crossing line-work. *Element joining* can assist in further reducing design file size and also in speeding future re-symbolization and attribution efforts (i.e., fewer elements to alter). *Duplicate removal* can be used with a variety of criteria settings to remove redundant data. Finally, the *dangle*

processor is primarily useful in identifying line-work errors, which were not corrected with the given tolerances. These flagged elements can then be efficiently visited and corrected using MRF Flag Editor. See Table 6.2 below for a summary of resource mapping recommendations.

Table 6.2. Modular summary for resource data processing.

MRF Clean Module	Recommended for Use?	Comments
Complex Element Stroking	Yes	Tolerance-based to preserve quality and improve efficiency.
Line Simplification	Yes	Improves efficiency based on known data quality levels.
Short Element Processing	Yes	Eliminates short spikes and digitizing errors.
Extension	No	May reduce the effectiveness of intersection creation.
Intersection Creation	Yes	Ensures connectivity and closure.
Duplicate Removal	Yes	Use criteria to refine element comparisons (i.e., database linkage review).
Element Joining	Yes	Reduces number of elements for future processing/analysis.
Conflation	Yes	Matches element geometry to remove sliver polygons.
Dangle Processing	Yes	Flags remaining errors for queued-editing with Flag Editor.

7. Learning MRF Clean with Example Design Files

You are encouraged to use the supplied example design files to understand the cleaning effects of MRF Clean.

File name	Cleaning Goals	Levels	Tolerance (MU)
Example1.dgn	Correcting overshoots and undershoots	1	30.0
Example2.dgn	Line weeding	1	20.0
Example3.dgn	Moving less accurate points to more accurate points.	1 2 3	20.0 5.0 90.0
Example4.dgn	Removing dangles	1	15.0
Example5.dgn	Removing duplicate and near-duplicate lines.	1	90.0
Example6.dgn	Moving less accurate points to more accurate points. Joining shorter elements to create longer elements	1 2	5.0 20.0
Example7.dgn	Joining shorter elements to create longer elements	1 2	20.0 5.0
Example8.dgn	Cleaning cells and lines	1 2	3.0 2.0
Example9.dgn	Cleaning arcs, cells and lines	1 2 3	4.0 0.01 3.0

MRF Flag Editor USER'S GUIDE

For MicroStation V8/XM

May 2009



© MRF Geosystems Corporation
Suite 200, 625 – 14th Street N.W.
Calgary, Alberta
Canada T2N 2A1

Tel: (403) 216-5515
Fax: (403) 216-5518
Email: mrfsales@mrf.com
WWW: <http://www.mrf.com>

MRF Flag Editor Operations Guide

MRF Flag Editor is designed to be used in conjunction with MRF Clean, MRF Clean 3D, and MRF Polygon to assist you in interactive 2D or 3D map editing. For example, when you run MRF Clean, the flagging options flag free end points with the character “D”, intersections with the character “X”, and sub-tolerance elements with the character “S” (see **Figure 1**). MRF Flag Editor is designed to drive you directly to these flags so you may review the linework and make editing changes to the linework.

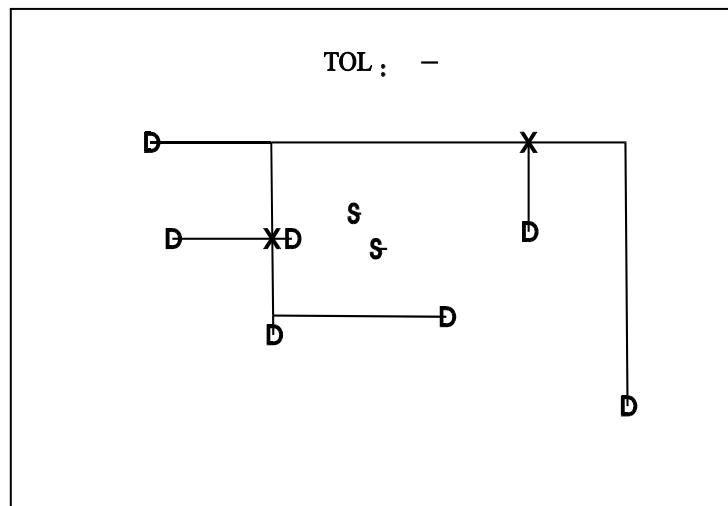


Figure 1: Flagged Linework

The following dialog box will be displayed after you load MRF Flag Editor by issuing “**mdl load mrfflag**” in the MicroStation command window:



Figure 2: MRF Flag Main Dialog Box

Defaults

The **Flag Type** option is set to “D”, the **Flag Level** is set to 63, and **Zoom Factor** is set to 0.2 by default. The **Edit Status** text field is set to “press next”. Only the push-buttons **Next >>** and **Help** are enabled. You can change **Flag Type**, **Flag Level**, and **Zoom Factor** before you start to edit the flags.

The first time the **Next >>** push-button is pressed, MRF Flag Editor will scan the file for flags.



Note: Only the current Flag_type on the current Flag_level will be scanned for.

The progress of the scanning process is displayed in the MicroStation command window.

After scanning is completed, MRF Flag Editor will

- display the first flag at the center of View 1,

- highlight the flag and the element that the flag is associated with, using the MicroStation current highlight color,
- display an editing status in the **Edit Status** field in the “MRF Flag Editor” dialog box, and
- enable the push-buttons for editing the flags.

You are now ready to edit the current flag.

Editing Tools

- Click the **Zoom In** and **Zoom Out** push-buttons to see more or less detail of the flag and its spatial relationship with neighboring elements. The current flag, and the element associated with the current flag are highlighted after the zooming operation. The “snap” point of the flag will be in the center of the window after you click **Zoom In** or **Zoom Out**.
- Change the **Zoom Factor** to set the zoom level. The best way to change the zoom factor is to hit the “Zoom In” or “Zoom Out” push-button. When you click the “Zoom In” or “Zoom Out” button, you will notice that the zooming factor will change accordingly. As soon as the **Zoom Factor** is changed, the current flag, if it exists, will be displayed using the new **Zoom Factor**. The flag will occupy **Zoom Factor** * 100% of your MicroStation Window. When you zoom-in or zoom-out to a certain level, and go to the NEXT flag or the PREVIOUS flag, the zooming factor will remain unchanged unless you click the “Zoom In” or “Zoom Out” button again.
- Click **Next >>** or << **Prev** to skip to the next or the previous flag.
- Click **Delete Flag** to delete the current flag and go to the next one.
- Click **Delete Flag+Elm** to delete both the flag and the element with which the flag is associated, and then go to the next flag.
- Click **Delete All Flags** to delete all the flags.
- Edit your design file using MicroStation tools.



Note: If you undo your edit to undelete the flags using MicroStation's "UNDO" tool, you should reload MRF Flag Editor to ensure that all the flags are re-scanned in to memory.

Reading the Edit_Status field

The **Edit Status** field normally displays a text in the format of **n/m**, where **m** is the total number of flags and **n** indicates that the current flag is the **n**th flag. If there are no flags left, the **Edit Status** field is set to "**done**".

If the current flag is the last flag ($n = m$), **Next >>** is disabled. If the current flag is the first flag ($n = 1$), **Prev <<** is disabled. If there is only one flag left ($m = 1$), only the **Next >>** push-button is enabled.

If $m > 0$, the push-buttons **Delete Flag** and **Delete All Flags** are enabled.

The push-button **Delete Flag+Elm** will be enabled when only one element associated with the current flag is found. It will be disabled if

- multiple elements end at, or are within 4 UORs of the position of the current flag,
- the current **Flag Type** is "X", or
- the element has been associated with a flag at the other end of the element.

If there is no flag to be edited ($m = 0$), MRF Flag Editor will:

- disable all the push-buttons, except **Next >>** and **Help**,
- set the **Edit Status** to "done",
- issue a beep, and
- display "Edit is done for your current flag type".

If you change the **Flag Type**, MRF Flag Editor will:

- disable all push-buttons except **Next >>** and **Help**,
- issue a beep, and
- display "**Flag Type** is changed. Press **Next >>** to start editing".

Once you press **Next >>**, MRF Flag Editor will re-scan the flags and the associated elements.

If you change **Flag Level**, MRF Flag Editor will:

- disable all push-buttons except **Next >>** and **Help**,
- issue a beep, and
- display “**Flag Level** has changed. Press **Next >>** to start editing”.

Once you press **Next >>**, MRF Flag Editor will re-scan the flags and the associated elements.

If you compress your current DGN file or add elements to the current DGN file, and click any push-button other than **Help**, MRF Flag Editor will:

- disable all push-buttons except **Next >>** and **Help**,
- issue a beep, and
- display “DGN file size is changed. Press **Next >>** to start editing”.

Once you press **Next >>**, MRF Flag Editor will re-scan the flags and the associated elements.

If you change to another DGN file, and click any push-button other than **Help**, MRF Flag Editor will:

- disable all push-buttons except **Next >>** and **Help**,
- issue a beep, and
- display “DGN file name is changed. Press **Next >>** to start editing”.

Once you press **Next >>**, MRF Flag Editor will re-scan the flags and the associated elements.



Note: You can click the “**Help**” button at any time to open an on-line version of the above operation guide.

You can learn more about using MRF Flag Editor with the supplied files FLAG2D.DGN and FLAG3D.DGN. These files contain existing flags for danglers, intersections and sub tolerance elements and will assist you in becoming familiar with the operation of MRF Flag Editor.

MRF Clean 3D *USER'S GUIDE*

For MicroStation V8/XM

May 2009



© MRF Geosystems Corporation
Suite 200, 625 – 14th Street N.W.
Calgary, Alberta
CANADA T2N 2A1

Tel: (403) 216-5515
Fax: (403) 216-5518
Email: mrfsales@mrf.com
WWW: <http://www.mrf.com>

Introduction

MRF Clean 3D is three-dimensional data cleaning software. It is written entirely in MicroStation Development Language (MDL) and runs anywhere MicroStation runs. It is a data cleaner designed to clean linear elements, unshared cells, or both.

MRF Clean 3D can:

- correct over-shooting
- correct under-shooting
- perform line weeding
- remove duplicate or near-duplicate points and lines
- create intersections in crossing linework to ease building of topologies
- remove or flag dangles
- join singly-connected line strings with identical attributes to form longer ones
- merge linework that has approximately the same geometry (conflation)
- identify free end points for manual review and editing
- reduce design file size to save disk space and increase display speed

A better alternative to single-tolerance data cleaning

Most GIS applications require the use of many data sets, which are collected by public agencies and private organizations, frequently using different positional accuracy. Also, in a GIS, the data sets were probably stored on different levels. Since many spatial features share common boundaries, such as a forest stand and a lake, these common boundaries must agree with each other even when they are stored on different levels.

Example: A forest company may use two types of maps: maps containing lakes, rivers and roads from the state or provincial government, and forest stand maps collected by the company. Assume that the topographic maps have a positional accuracy of two meters, and the forest stand maps have a positional accuracy of four meters. Let us also assume that the two map types are stored on different levels.

In cleaning these two types of maps, the forest company faces several challenges. If the two levels are cleaned separately, lakes and forest stands may overlap after cleaning, even though they should share a common

boundary. If the two levels are merged into one level and cleaned using a single tolerance of two meters, then most digitizing errors such as over-shooting and under-shooting in the linework from the forest stand map will not be corrected. And finally, if the merged level is cleaned using a single tolerance of four meters, the linework from the topographic map will be generalized.

MRF solution: Multi-tolerance Data Cleaning

Using multi-tolerance data cleaning, points with a larger tolerance can be moved a greater distance to coincide with points having a smaller tolerance. In this way, inconsistencies, such as the overlap or gap between a lake boundary and a forest stand boundary, can be corrected by moving the less accurate boundary—the forest stand—to the more accurate boundary of the lake. Also, when input points have identical tolerances, the movements of points is performed in such a way that straight lines are kept straight—a very desirable behavior for many GIS/LIS applications.

To the best of our knowledge, **MRF Clean 3D is the *ONLY* multi-level, multi-tolerance 3D data-cleaning package on the market.**

Other MRF Clean 3D features

MRF Clean 3D allows you to maintain inter-level geometric consistency. The following is an application where inter-level geometric consistency is desirable:

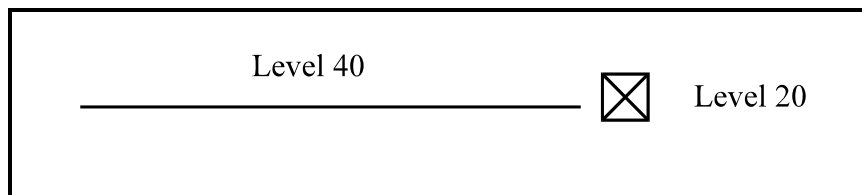


Figure 1: Geometric inconsistency

In property mapping, parcel boundaries on one level may not be geometrically consistent with corner markers on another level. For example, a lot boundary on level 40 may not end at the center of a cell on level 20 (see **Figure 1**).

A user might assign the following tolerances to the listed entities:

ENTITY	TOLERANCE (inch)
Survey pins	1
Plan boundaries	3
Block boundaries	5
Lot boundaries	10

Table 1: Sample Tolerances for property mapping

When cleaning the property map, MRF Clean 3D will move an entity with a larger tolerance to align with one with a smaller tolerance:

- plan boundaries will end at accurately placed survey pins
- block boundaries will coincide exactly with plan boundaries
- lot boundaries will coincide exactly with block boundaries

MRF Clean 3D is the result of many years of research and development. Extensive testing and user feedback indicate that MRF Clean 3D can improve the quality of data conversion and data editing for mapping, GIS, and AM/FM applications. At the same time, it can **increase productivity by TWO to TEN times.**

Elements Processed by MRF Clean 3D

MSTN Element Type	MSTN Element Number	MRF Clean 3D Processing
CELL_HEADER_ELM	2	cleaned as a point feature ¹ or line strings ²
LINE_ELM	3	converted to line
LINE_STRING_ELM	4	converted to line string
SHAPE_ELM	6	converted to line string
CURVE_ELM	11	stroked to line strings
CMPLX_STRING_ELM	12	stroked to line strings ³
CMPLX_SHAPE_ELM	14	stroked to line strings ³
ARC_ELM	16	stroked to line strings
ALL OTHER ELEMENTS		copied to output ⁴

Table 2: Elements processed by MRF Clean 3D

- Note 1:** When the cell is not a group hole, it is cleaned as a point feature. The cell header stores all the levels of its elements. For cleaning purposes the cell is treated as residing on its lowest level, which will be used to determine whether the cell should be cleaned and what tolerance should be used if the cell is to be cleaned.
- Note 2:** When the cell is a group hole, it is converted to a set of line strings. Their levels are set to the lowest level of the cell.
- Note 3:** Since each CMPLX_STRING_ELM or CMPLX_SHAPE_ELM has at least four words of attribute data, the attribute data will be copied to the stroked line strings.
- Note 4:** If no output file is specified, these elements will be left in the input file.

MRF Clean 3D Work Flow

The concepts presented in this section demonstrate how MRF Clean 3D goes about cleaning linework. The actual processing sequence depends on the parameters you specify. The figures presented in this section use line strings to demonstrate the principles employed by MRF Clean 3D. The same principles also apply to cells.

Reading Input Elements

All elements on user-selected levels (i.e. a level with non-negative tolerance) will be read in for cleaning. If a fence is defined, only the selected elements inside or overlapping the fence will be processed.



Note: Certain input elements are stroked into line strings. For more details about input elements, refer to Elements Processed by MRF Clean 3D.



Note: Each line string is simplified using the Douglas-Peucker algorithm.

Intersection

MRF Clean 3D recognizes two types of linework that may cause the creation of a true or a fuzzy intersections.

True intersection: When two vectors cross each other, the crossover point (X) is called a **true intersection** (see **Figure 2**).

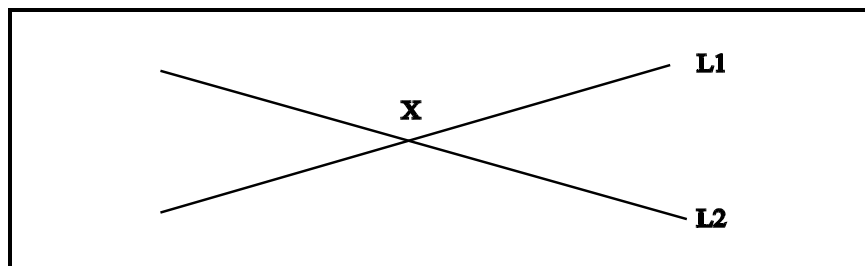


Figure 2: True intersection

Fuzzy intersection: A fuzzy intersection can be created from elements that are within one of the tolerance distances, but do not actually intersect with the existing linework. In other words, the two lines nearly intersect, and the distance from the end of one line to another line is shorter than one of the two lines' tolerances (see **Figure 3**). The **critical intersection** point on a line is created by the **end intersection** of a nearby line:

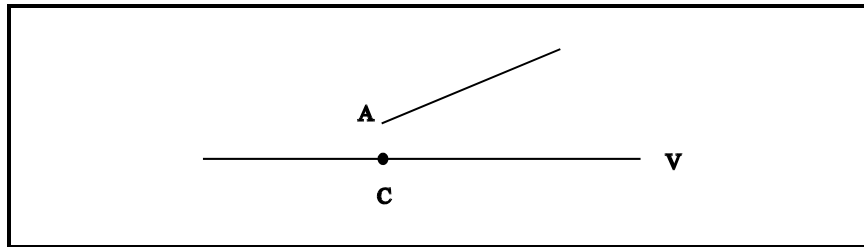


Figure 3: Fuzzy intersection

If the distance from end point A to line V is shorter than one of the two lines' tolerances, then node A is projected (shortest distance) to line V to determine where the intersection is actually located. The projected point (C) is called a **critical intersection** while A is called an **end intersection**.

Critical intersections are also created when the shortest distance between two lines (AB in **Figure 4**) is less than the tolerance of one of the lines:

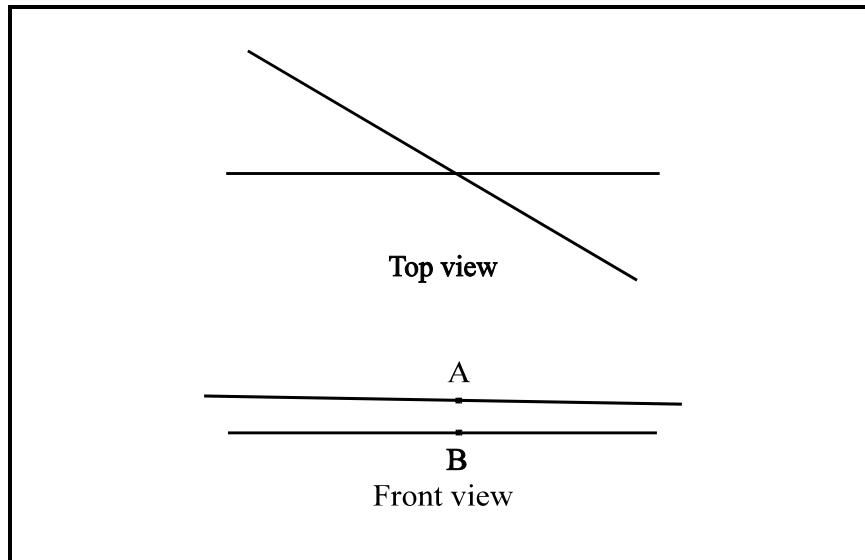


Figure 4: Critical intersection

Each intersection point inherits its tolerance from its input element. In **Figure 3**, intersection point A will have the same tolerance as the line that contains A, and intersection point C will have the same tolerance as line V.

Clustering

In this process, intersections are clustered together, or more specifically, intersections with larger tolerances are moved to intersections with smaller tolerances. However, no intersection is moved out of its tolerance. Among points with identical tolerances, a true intersection has a higher priority than a critical intersection, which in turn has a higher priority than an end intersection. **Figure 5** shows the result of clustering point C to point A when line V has a larger tolerance than point A. On the other hand, if line V has a tolerance smaller than that of point A, A will be clustered to point C (see **Figure 6**).

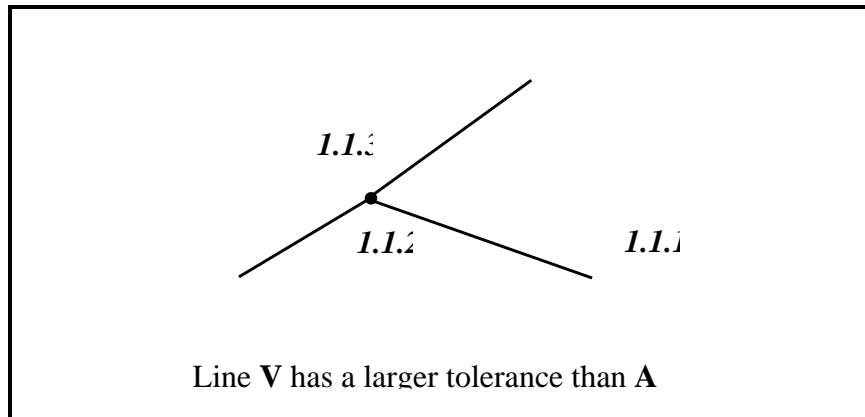


Figure 5: Intersection movement in clustering

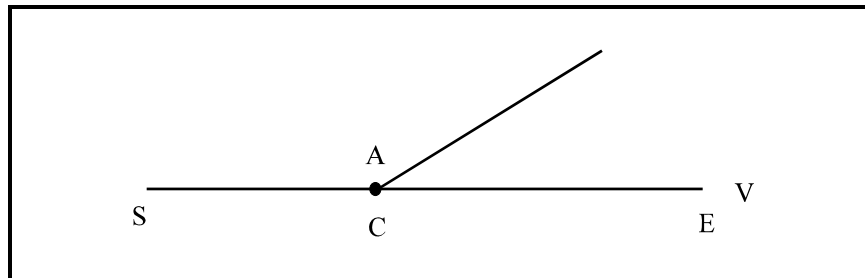


Figure 6: Clustering

Splitting

This function causes input line strings to split at their intersections to form a set of new line strings. For example, input line string V in **Figure 6**, will be split into two line strings SC and CE (assuming S and E are the start end points). Each new line string inherits the following attribute information from its corresponding input element:

- level
- graphic group number
- class
- property bits
- symbology (color, line weight, and line style)
- attribute linkages

Duplicate Removal

The intersection process may create exact-duplicate elements. The duplicate removal function removes such duplicates to reduce file size and to aid in topology building. In **Figure 7**, two input lines are within tolerance. As a result, end points A and D would create critical intersections at C and B. Assuming that line AB has a smaller tolerance, C will be clustered to A, and D will be clustered to B.

Since, after splitting, CD and AB will have identical geometry, they will become candidates for duplicate removal. If you select By Geometry for duplicate removal, all but one of the elements with the same geometry after clustering will be deleted. Only the element with the smaller tolerance will be kept. If both elements have the same tolerance, one selected at random will be retained and the other deleted. In the figure, CD will be removed because it has a larger tolerance.

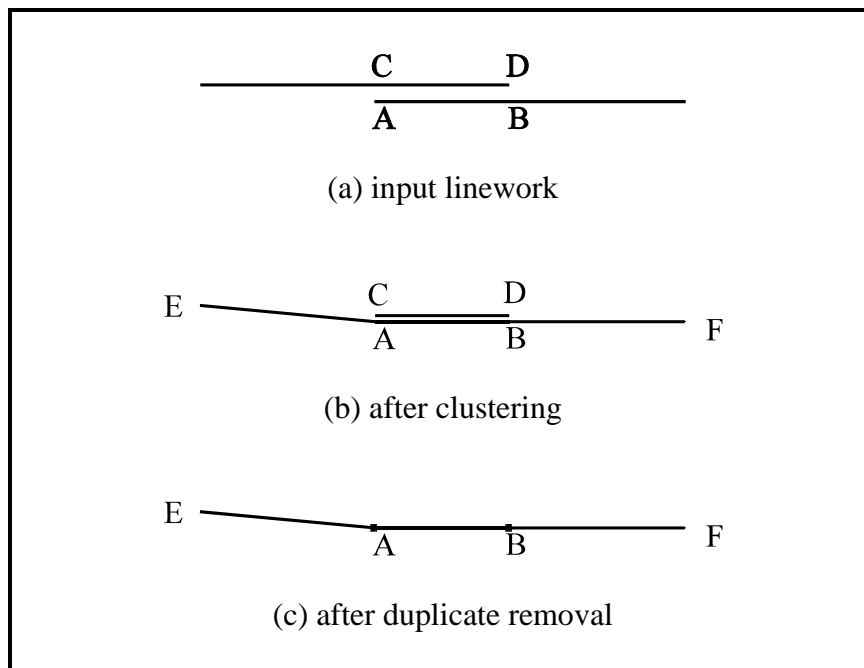


Figure 7: Duplicate removal

If you choose to remove duplicates BY ATTRIBUTE, AB and CD will be considered to be duplicates if they have identical geometry after clustering, and have the same values for the following MicroStation element properties:

- level
- graphic group number
- class
- property bits
- symbology (color, line weight, and line style)
- attribute linkages.



Note: When checking attribute linkages, MRF Clean 3D will ignore the **information, remote and modified linkage property bits.**



CAUTION: When in doubt choose the "BY ATTRIBUTE" option to preserve duplicate lines. If you want to remove all the duplicate or near duplicate linework, turn on the Fuzzy Intersection and True Intersection options.

Element Joining

If just two line strings are singly connected at a node or end point, they will be joined to form one longer line string, if they have the same values for:

- level
- graphic group number
- class
- property bits
- symbology (color, line weight, and line style)
- attribute linkages.



Note: When checking database linkages, MRF Clean 3D will ignore the **information,**

remote and modified attribute linkage property bits.

In **Figure 8**, assume that the input lines have identical attributes. After duplicate removal, we will be left with three line strings: EC, CB, and BF. After element joining, we will have one line string with four points: E, C, B, and F.

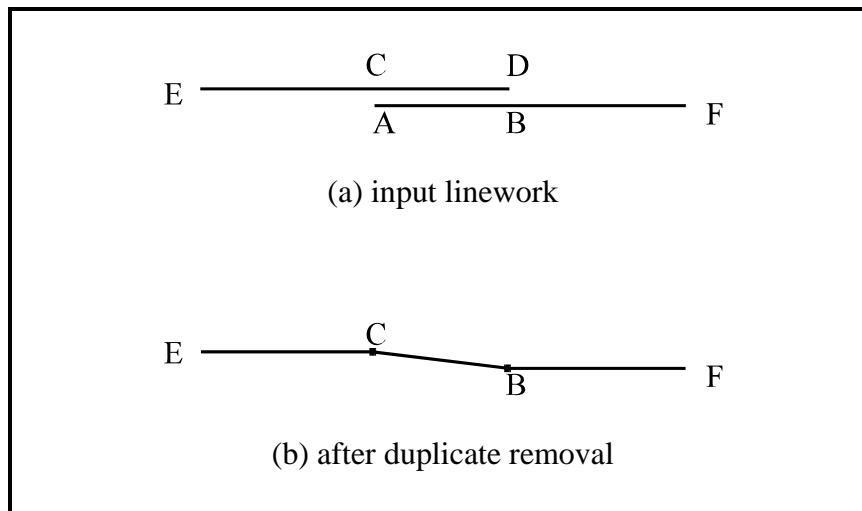


Figure 8: Element joining

If there are multiple elements with identical geometry but different attributes, element joining is still carried out as long as network topology is maintained. For example, if there are elements on three different levels as shown in **Figure 9**, we will have three line strings with the points E, C, and B after element joining—assuming that the elements on each level can be joined.

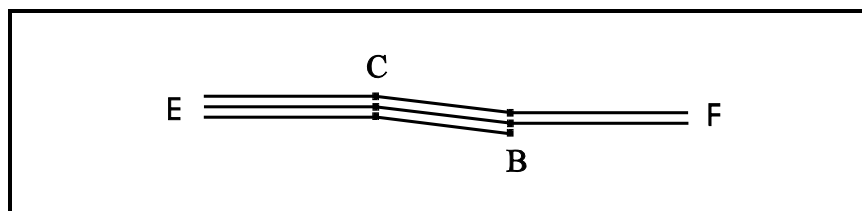


Figure 9: Multi-line joining

Dangle Removal

A dangle is a linear element that has at least one end that is not attached to another linear element or point. A dangle with a length less than (**Dangle Factor** * tolerance) will be removed by default. Also by default, the location of dangles which are longer than (**Dangle Factor** * tolerance) will be marked with the text element letter “D” placed on the user specified **Free End Flag Level**.

Conflation

Conflation is the merging of linework that has the same or approximately the same geometry. It removes slivers and gaps, and aligns linear elements that should have the same geometry—such as national, state, and city political boundaries.



Note: Two line string elements are conflated together if they share the same start and end nodes, and are within tolerance.

After conflation, the line string with a larger tolerance will have the same geometry as the line string with the smaller tolerance. In **Figure 10**, line strings AB and ACB share the same nodes, A and B. If AB has a smaller tolerance, and ACB can be conflated to AB, then ACB will have the same geometry as AB after conflation. If AB has a larger tolerance and AB can be conflated to ACB, AB will have the same geometry as ACB after conflation. After conflation, exact duplicates will be removed as explained in sub-section *Duplicate Removal*.

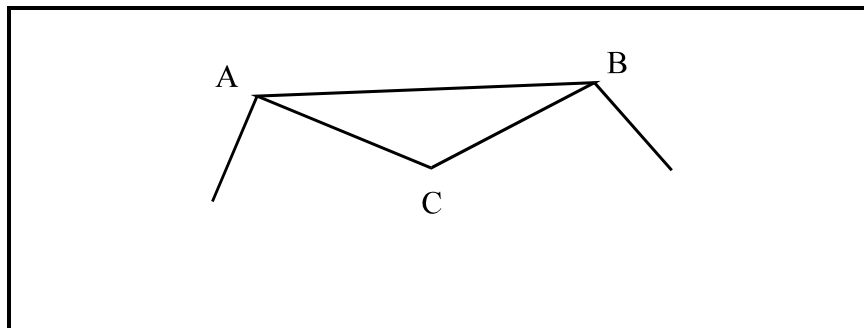


Figure 10: Conflation

Output

All the cleaned elements (cells and line strings) are written to the output file.

MRF Clean 3D GUI

MRF Clean 3D parameters can be entered by means of a parameter file—optional for interactive processing but required for batch processing—or interactively on the **MRF Clean 3D** and **MRF Clean 3D Parameters** forms. For more information about interactive and batch processing, refer to *Running MRF Clean 3D Interactively* and *Batch Processing*. This section discusses how each parameter affects MRF Clean 3D processing.

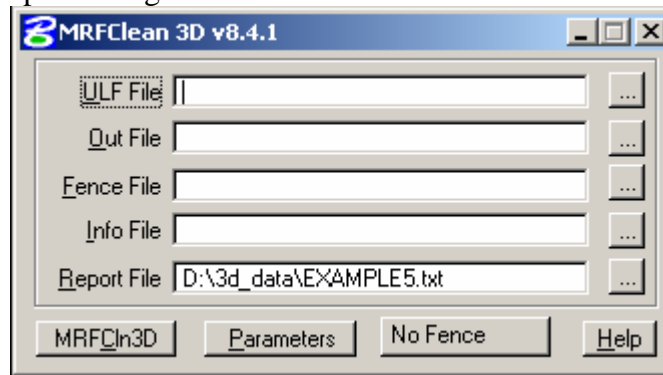


Figure 11: MRF Clean 3D Main Dialog Box

MRF Clean 3D Input Parameters

ULF File: (This field is optional. By default, the currently active design file is the input). When a ULF file is specified, MRF Clean 3D will open the design file specified in the ULF file. The elements listed in the ULF file will be cleaned using the user-specified tolerances and parameters. All other elements in the input design file will be copied to output. If you have drawn a fence, only the elements inside or overlapping the fence will be processed.



Note: You can activate the file picker beside the ULF File field to select a ULF file.

Out File: (This field is optional. By default, the currently active design file is the output file). If an output file is specified, MRF Clean 3D will copy to it:

- all elements not listed in the ULF file,
- all elements on levels that will not be processed,
- all elements not handled by MRF Clean 3D, and
- the results of the cleaning process.



Note: You can use the file picker beside the Out DGN File field to select an existing directory for the file.



Note: The cleaned design file can be edited using MicroStation editing tools, and cleaned again.



Recommendation: It is good idea to always specify an output file, and keep the original design file intact. If you specify a design file that already exists, an alert dialog box will pop up and ask for permission to overwrite. In batch mode, MRF Clean 3D will not overwrite an existing file.

Fence Option: (Default is **No Fence**)

There are four settings available:

1. No Fence

MRF Clean 3D will process the entire design file. Use this option when the input file is small.

2. Use Current

MRF Clean 3D will clean the elements inside or overlapping the current fence. Use the MicroStation “place fence block” or “place fence shape” tool to define a fence before selecting this option. Use

this option when you want to restrict processing to a small area of the file. This setting is not available in batch mode.

3. Use Fence File

If you select this option, a file selector will pop up, requesting the **In Fence File** name. You should select an existing DGN file containing a set of shapes. If you press “Cancel” in the file selector dialog, the fence option will re-set to **No Fence**. Once a valid fence file is specified, MRF Clean 3D will clean the design file multiple times using each shape element within the fence file as a separate fence. The fence file should have the same settings as the file to be cleaned. Use this option when only a few areas need to be cleaned.

4. Auto Partition

If you select this option, a dialog box will pop up, show the input DGN file size, and ask you to enter a partition size. The partition size will determine how many fences MRF Clean 3D will create.



Note: The number of fences is obtained by dividing input file size by partition size.

Both the input file size and the partition size are in megabytes (MB). For example, if your input size is 5 MB and your partition size is 0.5 MB, MRF Clean 3D will create about 10 fences depending on the range rectangle of your input file. In **Figure 12**, since each fence is a square, you may get 12 (3 by 4), 10 (5 by 2) or 9 (3 by 3) fences, depending on the range rectangle.

9	10	11	12
5	6	7	8
1	2	3	4

Figure 12: Twelve fences (4 by 3) placed by MRF Clean 3D with **Auto Partition** option

To write these fences out to a design file, specify an **Out Fence File** (see the following parameter). The **Out Fence File** should be an

existing DGN file (preferably an empty DGN file) with the same settings as the file to be cleaned.



Recommendation: Use the AUTO_PARTITION option when you want to clean an entire large file. This option will deliver good performance and reduce overall memory requirements.

After the successful completion of the cleaning process, MRF Clean 3D resets this option to **No Fence**. The MRF Clean 3D report file (see **Report File**) will have one line indicating if a fence is used in the cleaning process.

Out Fence File: (This field is optional: by default no fences will be recorded. For more information about fences, see **Use Fence**). The **Out Fence File** keeps a record of all the fences that have been used in previous cleaning processes. You can reference the **Out Fence File** before drawing the next fence to avoid missing a certain area or processing an area multiple times. The **Out Fence File** can also be used as a reference when cleaning other overlays covering the same area.



Note: The 'Out Fence File' must be an existing file. MRF Clean 3D will not create one if it does not already exist.

There are two ways to create an **Out Fence File**:

1. make a copy of the file to be cleaned and delete the contents, or
2. copy the same seed file used to create the file to be cleaned

You should make sure that the specified file has the correct settings, such as working units. If you do not specify an existing file, or no fence has been defined, no fences will be written.



Note: You can activate the file picker by pushing the button beside the 'Out Fence File' field.

Info File: (Optional, ASCII text file). This file contains the parameters for running MRF Clean 3D when you want to ensure that the same set of parameters are applied to a set of files. Toggle button values are represented by 0 for **OFF** or 1 for **ON**. An option button can have as many values as the number of options: the first option is 1, the second 2, etc. The ordering of the options is the same as displayed in the dialog box.

The info file has the following format:

```
FENCE_OPTION:1
IN_FENCE_FILE:
AUTO_PARTITION_SIZE:0.200000
CURVE_FACTOR:2.000000
DANGLE_FACTOR:1.000000
FILTER_FACTOR:1.000000
FREE_END_FLAG_LEVEL:63
INTERSECTION_FLAG_LEVEL:
SUB_TOL_ELE_FLAG_LEVEL:
FLAG_FONT_NUMBER:3
REMOVE_DUPLICATES_BY_ATTRIBUTE_OR_GEOMETRY:1
USE_CELL_AS_NODE_OR_NON_NODE:1
BY_LEVEL:0
FAST:0
FUZZY_INTERSECTION:1
TRUE_INTERSECTION:1
DEL_SUB_TOL_ELE:1
LEVEL_1:1.000000
```

List 1: Info file format

Note that while the format of the info file is important, the ordering of the parameters is not. All uppercase characters before “:” on each line in the above info file template are considered to be keywords (keywords are generally the same as the parameter captions on the various dialog boxes, but all uppercase with spaces replaced by underscores). No space should appear before a keyword. If a keyword appears in the info file more than once, the last appearance of the keyword will be used to set the value. Each parameter should occupy one line only. You can add comments as shown below:

```
FENCE_OPTION:1;this is to set a fence_option
IN_FENCE_FILE:
The following line is to give a partition size
AUTO_PARTITION_SIZE:0.200000
CURVE_FACTOR:2.000000
DANGLE_FACTOR:1.000000
FILTER_FACTOR:1.000000
FREE_END_FLAG_LEVEL:63
INTERSECTION_FLAG_LEVEL:
SUB_TOL_ELE_FLAG_LEVEL:
FLAG_FONT_NUMBER:3
REMOVE_DUPLICATES_BY_ATTRIBUTE_OR_GEOMETRY:1
USE_CELL_AS_NODE_OR_NON_NODE:1
BY_LEVEL:0
FAST:0
FUZZY_INTERSECTION:1
TRUE_INTERSECTION:1
DEL_SUB_TOL_ELE:1
LEVEL_1:1.000000
```

List 2: Sample Info file

In the above example, both “*;this is to set a fence_option*” and “*The following line is to give a partition size*” are comments. In the first instance, the semicolon (“;”) is used as a comment separator. A comment may occupy a line by itself provided that it not begin with a keyword, as in the second instance. Parameters not specified in the info file are set to defaults.



Note: You can activate the file picker beside the 'Info File' field to select a File.



Note: Once an info file is specified, MRF Clean 3D will load the parameters from the info file and overwrite existing parameters.

Report File: (Optional, ASCII text file: by default MRF Clean 3D will use a file with the name of the first eight characters of the design file's name and the extension .TXT). MRF Clean 3D uses this file to log details of a cleaning process and echo the input parameters. The

default file name is the first eight characters of the input DGN file name with a .TXT extension. For example, if your input file is C:\DGN\MRFTEST.DGN, then the default report file is C:\DGN\MRFTEST.TXT. If no report file is specified, MRF Clean 3D uses MRFCLN3D.TXT in the current directory as the default. If the report file does not exist, one will be created.

The following information is appended to the report file:

- ◆ input file name
- ◆ input file size in bytes
- ◆ input parameters
- ◆ major MRF Clean 3D processes
- ◆ output file name
- ◆ output file size in bytes
- ◆ total number of linear elements processed
- ◆ number of true intersections fixed or flagged
- ◆ number of sub tolerance elements fixed or flagged
- ◆ number of free ends flagged
- ◆ completion or error message
- ◆ elapsed time of the cleaning operation

You can read the report file periodically to monitor processing in progress. The report file also helps you find out where MRF Clean 3D encountered problems.



Note: The report file is a prime source of information if you ever have a problem cleaning a design file.



Note: You can activate the file picker beside the 'Report File' to select an existing directory and enter a file name.

Parameters: When this button is pushed, MRF Clean 3D's parameter sub-dialog box will be opened. Parameter values are read in from the info file (if designated) or set to defaults.



Note: You can accept the default values or make changes in the parameter sub-dialog box as desired.



Note: Once the parameters sub-dialog box is opened, the parameters in this sub-dialog box will be used in the cleaning process.

Help: When this button is pushed, an on-line User's Manual will be displayed. You can scroll through the box for information about setting parameters.

MRFCl3D: When this button is pushed, MRF Clean 3D will start processing. If you want to abort MRF Clean 3D while it is processing, you can press CTRL + C (“^C”). When aborting MRF Clean 3D under WINDOWS NT or WINDOWS 95, an “Alert” window will open. You should click the “OK” button. At this point, a “MicroStation” text window will open. Simply minimize this window and carry on.

MRF Clean 3D Parameters

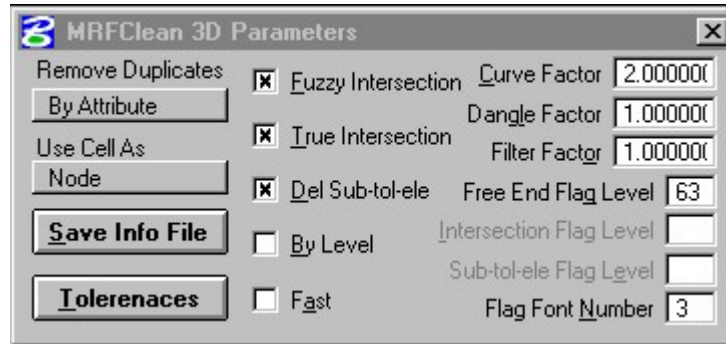


Figure 13: MRF Clean3D Parameters Dialog box

Remove Duplicates: (Default is By Attribute.) Duplicate and near duplicate elements from the input may become exact duplicates after splitting. Two options are available to remove these duplicates:

1. **By Attribute**

When this option is selected, MRF Clean 3D will only remove duplicate elements that have identical geometry and MicroStation element characteristics (refer to *MRF Clean 3D workflow*).

2. **By Geometry**

With this option activated, MRF Clean 3D will remove duplicate elements if they share the same geometry, regardless of their attributes. The duplicate element with the smallest tolerance will be kept. If a number of elements have the same tolerance, only one of them will be kept.

Use Cell As: (Default is Node.) This option has two settings: node and non-node. Even though a cell is a point object, it may be treated as a node, or line end, for various purposes, such as element joining, dangle removal, conflation, and free end flagging.



Note: This function works on both types of MicroStation cells: "point" and "graphic".

1. Node

MRF Clean 3D treats each cell as a node during element joining, dangle removal, conflation, and free end flagging. In some utility workflow, a line ending at a cell is not considered to be a dangle. For these situations the cell should be treated as a node.

2. Non-node

MRF Clean 3D will ignore cells in element joining, dangle removal, conflation, and free end flagging.

In **Figure 14**, if the cell at point C is treated as a node, line AC will not be considered a dangle. However, if the cell at point C is not treated as a node, then line AC will be a dangle.

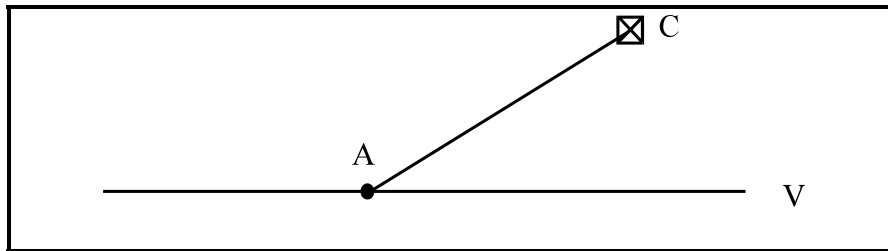


Figure 14: Use Cell option

Fuzzy Intersection: (Default is enabled.) If enabled, MRF Clean 3D will create intersections where linework is very close to intersecting, but not actually touching. Fuzzy intersections can be used to correct for undershoots and overshoots.



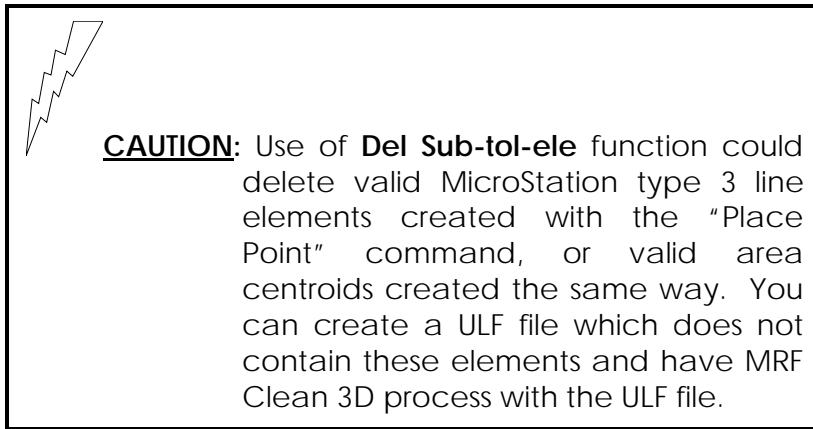
Note: MRF Clean 3D does not flag fuzzy intersections.

True Intersection: (Default is enabled.) If enabled, true intersections will be created (i.e. linework will be broken at the point where it crosses, and separate linear elements will be created). In addition, flagging of intersections and the **Intersection Flag Level** field will be disabled.

If you would prefer to flag all crossing linework—rather than create actual intersections—disable **True Intersection**, and enter a valid level number in the **Intersection Flag Level** field.

Del Sub-tol-ele: (Delete Sub-Tolerance Elements): (Default is disabled.) If enabled, any linear element that has a total length less than or equal to its tolerance will be deleted.

If you would prefer to flag all linework that is shorter than its tolerance—rather than delete it—disable **Del Sub-tol-ele**, and enter a valid level number in the **Sub-tol-ele Flag Level** field.



By Level: (Default is disabled) If enabled (specify 1 in info file), each selected level in the design file will be cleaned independently. Linework on one level will not intersect with that on other levels. When disabled (specify 0 in info file), all selected levels will be cleaned together.



Note: If you want to clean related linework on different levels, 'By Level' must be disabled.

Fast: (Default is disabled.) If enabled, dangle removal and conflation will not be performed. This reduces the time it takes to clean a large design file. Enable this option if you do not need these processes.

Curve Factor: (Default is 2.0) Curves, complex shapes, circular arcs, and complex chains are approximated by vectors with a stroking distance of (**Curve Factor** * tolerance). The stroking distance is the maximum distance between the curve and the approximating vectors. The **Curve Factor** field has a default value of 2.0, which means that the selected

curves will be stroked into line strings using $(2.0 * \text{tolerance})$ as the stroking distance. You can specify a value for the curve factor; its minimum value is 0.01.

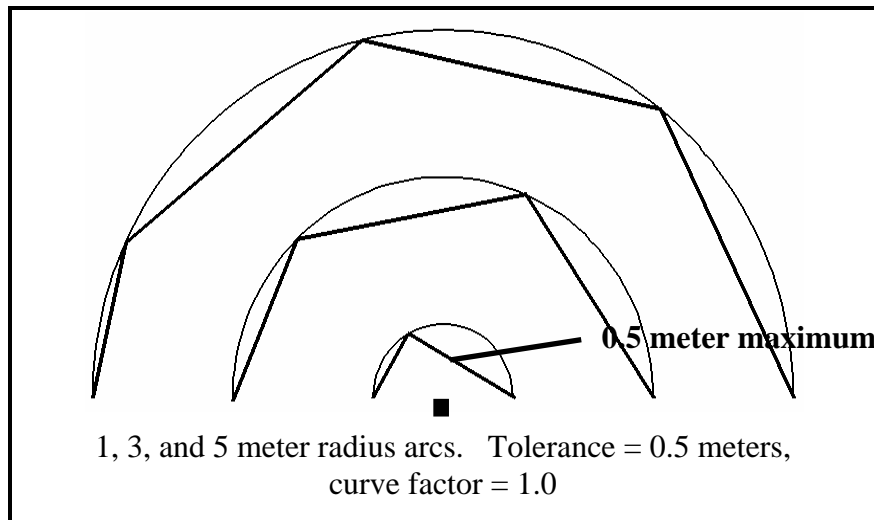


Figure 15: Arc stroking tolerance

The degree of smoothness of the stroking is a function of the curve factor, the tolerance, and the radius of the curve (see **Figure 15**). Smaller curve factors will always produce smoother stroked line strings, but will also produce larger elements with more vertices. Experiment with your data to develop the best set of parameters.

Dangle Factor: (Default is 1.0) A dangle is an element that has at least one free end point. MRF Clean 3D removes a dangle if its length is less than $(\text{Dangle Factor} * \text{tolerance})$. The minimum value for **Dangle Factor** is 0.0. This principally affects removal of overshoots. Some dangles, such as street centerlines ending at a cul-de-sac, may be appropriate to keep.

Filter Factor: (Default is 1.0) MRF Clean 3D allows you to weed the input elements (i.e. to simplify linework by removing unnecessary vertices). MRF Clean 3D uses a modified version of the Douglas-Peucker algorithm for this process. The modification produces better results for some special cases. This process uses a weeding tolerance of $(\text{Filter Factor} * \text{tolerance})$. The minimum value for **Filter Factor** is 0.0. The larger the value of $(\text{Filter Factor} * \text{tolerance})$, the more vertices will be removed. Here again, experiment with your data to develop the best set of parameters.

Free End Flag Level: (Optional. Default is level 63). If you specify a valid level in this field, MRF Clean 3D will place a text element “D” on that level to mark the location of each free end that could not be resolved as an overshoot or undershoot. The text is created using the current MicroStation settings for text size, weight, and justification. The font of the text is specified by **Flag Font Number**. Another MRF product, MRF Flag Editor, can be used to edit the marked free ends very efficiently. If a valid level is not specified, flags will not be placed.



Recommendation: Specify a level that contains no graphic elements. Also, setting the justification to Left Bottom will cause the lower left end of the “D” to align with the free end of the linear element. This will make it easier to distinguish the flagged element from linework that has been cleaned, or appears in a reference file.



TIP: The best set of parameters for flagging, and not removing dangles is:

Dangle Factor set to 0.0

Free End Flag Level set to a valid level

Intersection Flag Level: (Optional. Default is level 62 when True Intersection is disabled). If you specify a valid level in this field, MRF Clean 3D will place a text element “X” on that level to mark the location where each true intersection would normally be created. The text is created using the current MicroStation settings for text size, weight, and justification. The font of the text is specified by Flag Font Number. MRF Flag Editor can be used to locate the marked intersections very efficiently. If a valid level is not specified, flags will not be placed.



Recommendation: Specify a level that contains no graphic elements. Also, setting the

justification to Left Bottom will cause the lower left leg of the "X" to align with the point where the linear elements cross. This will make it easier to distinguish the flagged element from linework that has not have been cleaned, or appears in a reference file



TIP: The best set of parameters to use for flagging, and not breaking linework at true intersections is:

toggle True Intersection to OFF

set Intersection Flag Level to a valid level

toggle Fuzzy Intersection to OFF

set Free End Flag Level to blank

Sub-tol-ele Flag Level: (Optional. Default is level 61 when **Del Sub-tol-ele** is disabled). If you specify a valid level in this field, MRF Clean 3D will place a text element "S" on that level to mark the start point of each sub-tolerance element. The text is created using the current MicroStation settings for text size, weight, and justification. The font of the text is specified by **Flag Font Number**. MRF Flag Editor can be used to locate the marked sub-tolerance elements very efficiently. If a valid level is not specified, flags will not be placed.



Recommendation: Specify a level that contains no graphic elements. Also, setting the justification to Left Bottom will cause the lower left curl of the "S" to be near the start point of the linear element. This will make it easier to distinguish the flagged element from other linework that has not been cleaned, or appears in a reference file.



TIP: The best set of parameters to set for flagging, and not removing, short or sub tolerance elements is:

Del Sub-tol-ele set to OFF

Sub-tol-ele Flag Level set to a valid level

Flag Font Number: [Default is 3 (engineering)] This is the font used for flag characters. This parameter can be specified in the info file as well. You can always check for the fonts available within the system by clicking the “Text” item under the “Element” menu in MicroStation.

Save Info File: When this button is pushed, MRF Clean 3D will pop up a file creation dialog box and ask you to key in a file name. All the current parameter settings will be saved to the specified file. If the info file already exists, it will ask you whether to overwrite it or specify a different file name.



Recommendation: One work flow for cleaning a large number of files is to use a fence and clean a small portion of one file until the best set of parameters or sets of parameters (for sequential processing) are found. You can save these parameters into a file using **Save Info File**.

Tolerances: Push this button to bring up a dialog box to set the tolerance for each level with elements.

Sample Results of MRF Clean 3D

MRF Clean 3D uses generic algorithms to perform data cleaning and does not follow a set of pre-defined cases. The best way to discover the behavior of MRF Clean 3D is to construct test cases and process them. The following diagrams demonstrate the cleaning principles used. Each element is assumed to be on a unique level unless otherwise stated. Tolerances are shown at the top of each figure.

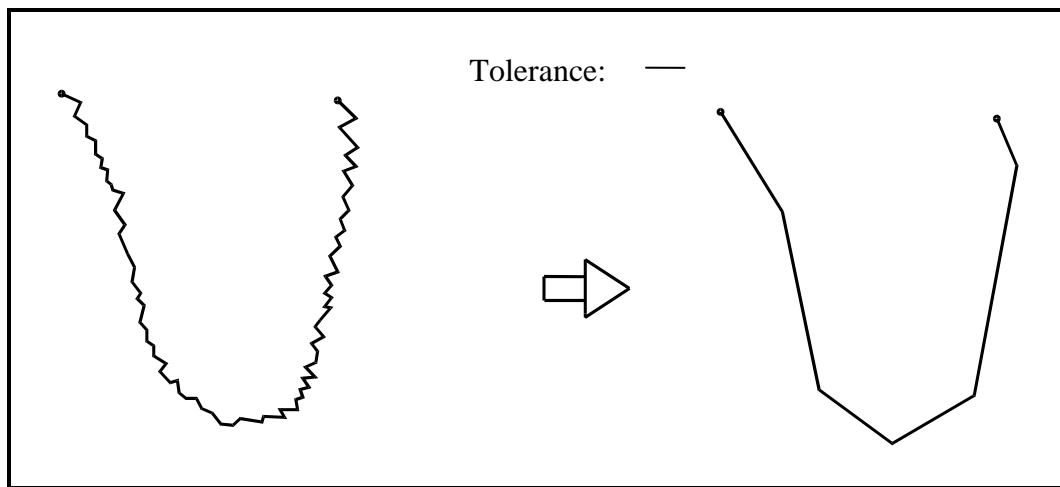


Figure 16: Redundant vertices removed by line weeding

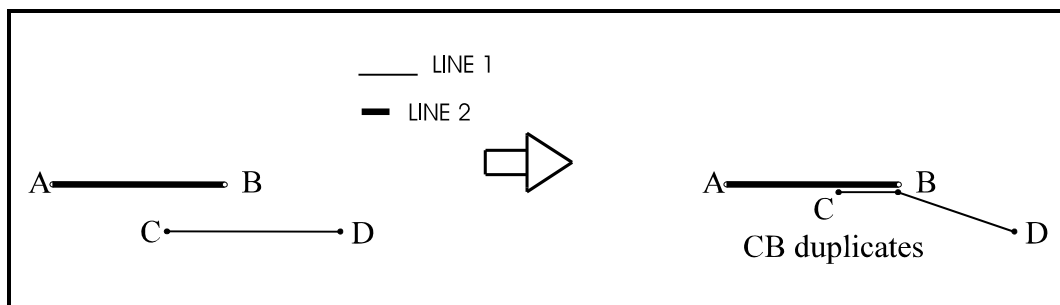


Figure 17: Line with larger tolerance moved to line with smaller tolerance

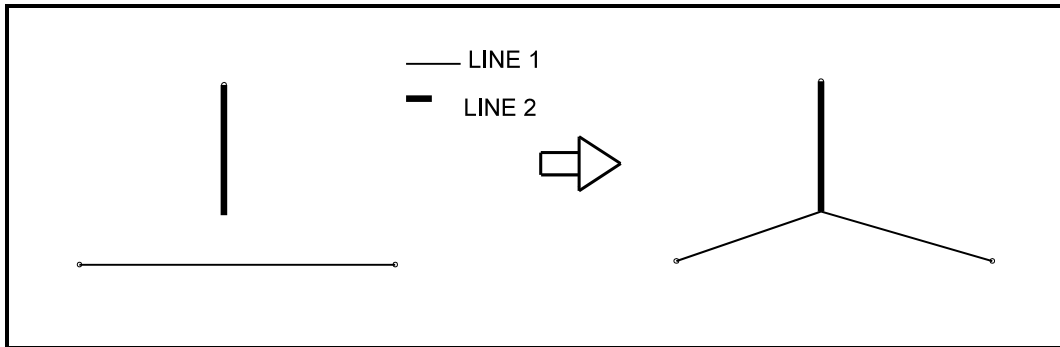


Figure 18: Fuzzy intersection created in linear element with larger tolerance

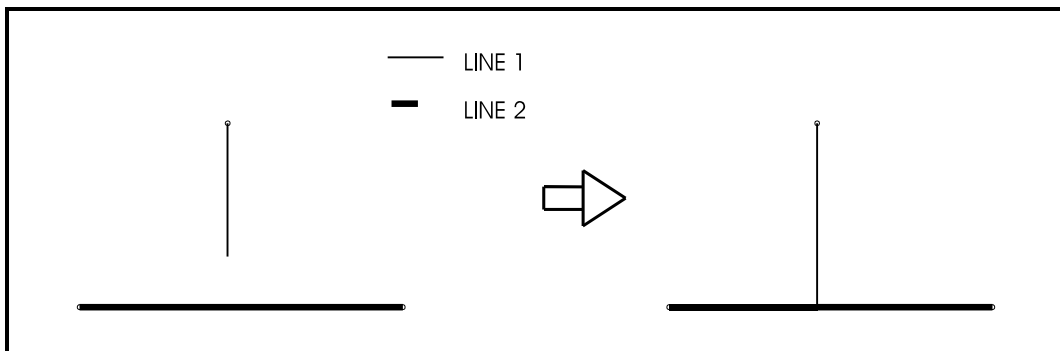


Figure 19: Fuzzy intersection created in linear element with smaller tolerance

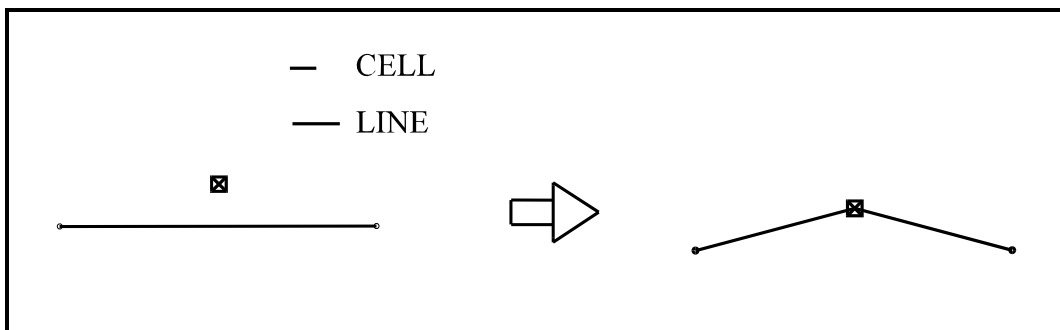


Figure 20: Fuzzy intersection created in linear element with larger tolerance

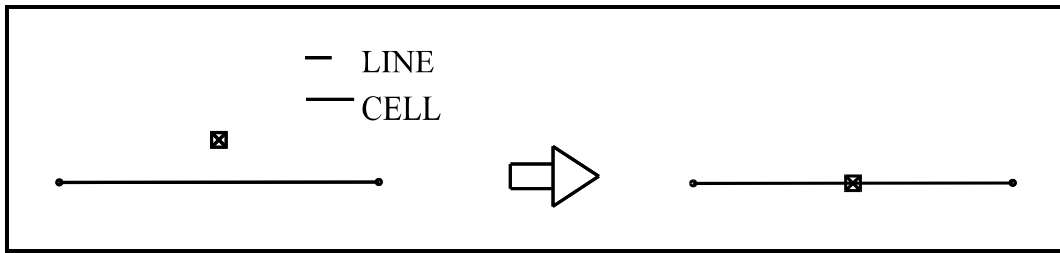


Figure 21: Cell with large tolerance moved to fuzzy intersection in linear element

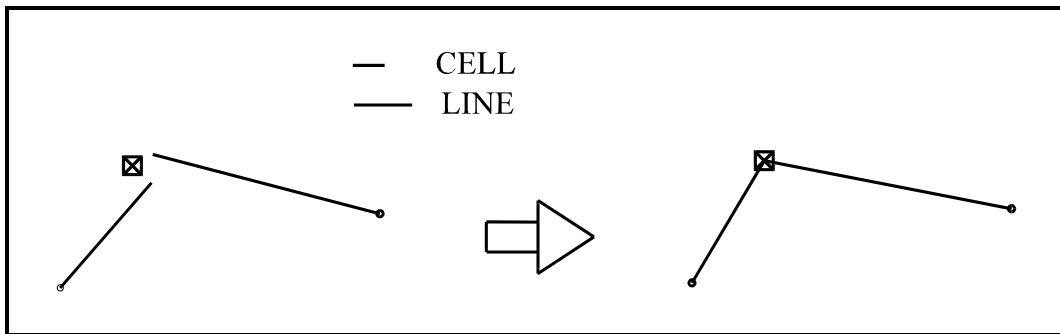


Figure 22: Linear elements with large tolerance moved to collocate with cell origin with smaller tolerance

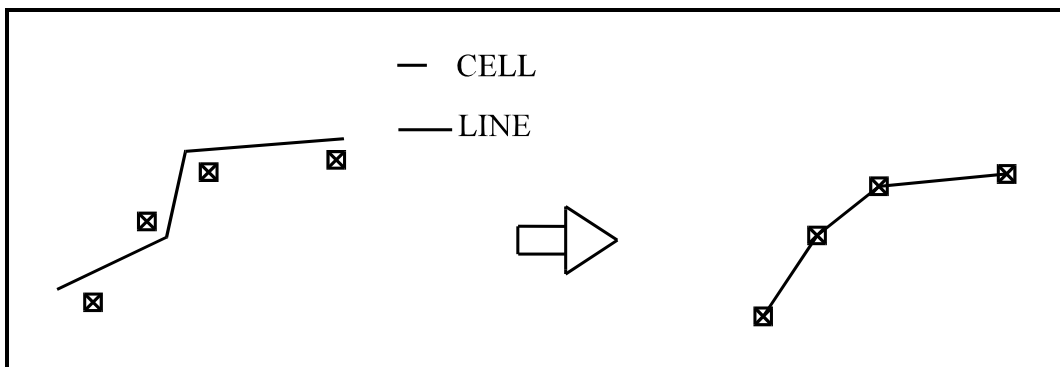


Figure 23: Nodes and vertices on linear element with large tolerance collocated at cell origins with smaller tolerances

Running MRF Clean 3D Interactively

Follow these steps to load and run MRF Clean 3D interactively:

- (1) Start MicroStation and open the design file to be cleaned.
- (2) Choose “MDL Applications” from the “User” menu.
- (3) Click on “mrfcln3d.ma.”
- (4) Click “LOAD”. MRF Clean 3D is now loaded. A dialog box will pop up.

⇒ **Tip:** Instead of using Steps 2 to 4 to load MRF Clean 3D, you can key in on the MicroStation Command Line:

mdl load mrfcln3d

The MRF Clean 3D dialog box will be displayed:

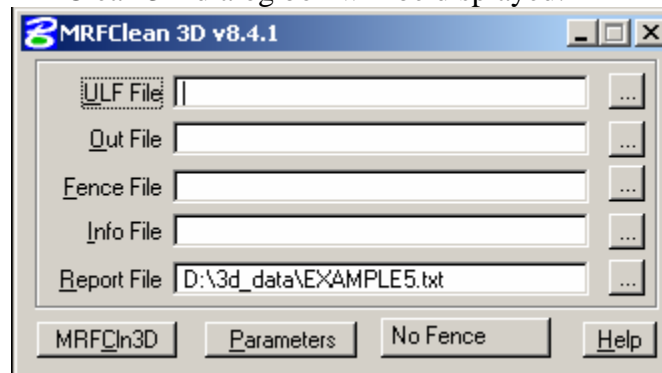


Figure 24: MRF Clean 3D Main Dialog Box

- (5) (Optional) Specify a ULF file in the **ULF File** text field, if you want to clean using ULF file. If a ULF file is not specified, MRF Clean 3D will process the currently active design file. Alternatively, you can activate the file picker beside the **ULF File** field to select a ULF file.
- (6) Specify the full path for the output design file in the **Out DGN File** text field. If the specified file already exists, a warning will be given and you will be asked for permission to

overwrite. If an output file is not specified, the cleaned elements will be written back to the current design file. If you do not specify the full path, MRF Clean 3D will create the output file in the current directory. Alternatively you can activate the file picker beside the **Out DGN File** field to select an existing directory and enter a file name.

- (7) (Optional) Specify the full path for the **Out Fence File**. If input is from a ULF file, the fence option will allow you to process the entities that are inside or overlapping the fence. If no fence is used go to step (9). If you do not specify the full path, MRF Clean 3D will assume the output fence file to be in the current directory. Alternatively you can activate the file picker beside the **Out Fence File** field to select a file.
- (8) (Optional) Specify the **Info File** containing parameters for running MRF Clean 3D. Alternatively you can activate the file picker beside the **Info File** field to select an info file. If you have such a file, you may go directly to step 11 to run MRF Clean 3D, or you may go to the next step to modify the parameters read from the info file. If you do not have an info file, go directly to step (9).
- (9) Choose a fence option (see **Fence Option**). Before selecting **Use Current** fence option, you should use MicroStation “place fence block” or “place fence shape” tool to digitize a fence in active design file. Then you can set **Fence Option** to **Use Current** to perform cleaning by fence in the active design file.
- (10) Click the **Parameters** button to open a sub-dialog box for entering/modifying parameters. The MRF Clean 3D Parameters form will appear.

Enter the **Curve Factor**, **Dangle Factor**, and other parameters as described previously.

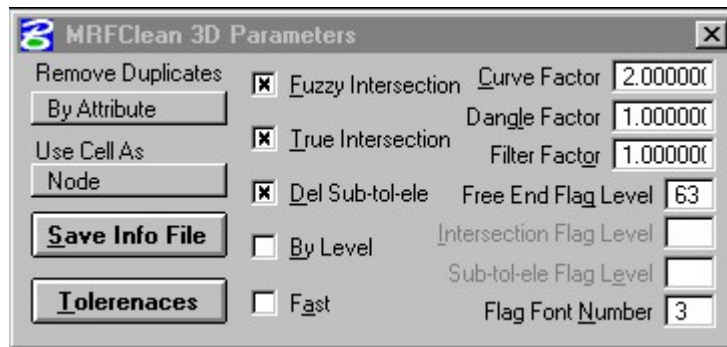


Figure 25: MRF Clean 3D Parameters Dialog Box



Note: If you have a question about any parameter or about the meaning of any terms used on the forms, click the Help button on the MRF Clean 3D form. A scrolling list of information will be displayed which describes the parameters in detail.

- (11) Press the **Tolerances** button and, in the dialog that appears, enter and/or modify the tolerance for each level requiring processing. To set the same tolerance for all levels, use the **Set All** option. To set individual tolerances, use the **Set** button.



Note: Use the "Measuring" tool from the MicroStation "Palettes" pull-down menu to measure the typical over-shoot or under-shoot distance for each level to be cleaned. Enter the distances (in Master units as displayed on the form) in the tolerance fields designated for the levels. If a level contains graphic elements that should not be processed, enter the tolerance value -0.01 for that level—levels with negative tolerance values are not processed.

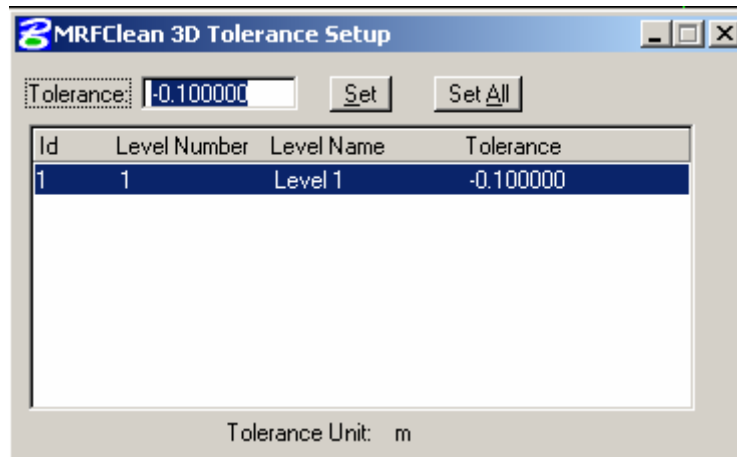


Figure 26: MRF Clean 3D Tolerances Dialog Box

- (12) Click the **MRFCln3D** button to start the cleaning process.

If one or more input parameters are incorrect, MRF Clean 3D will stop processing and display an error message (e.g., “output file already exists”) in the “Error” field of the MicroStation Command Window.



Note: If you wish to stop MRF Clean 3D in the middle of processing, press **CTRL+C** and MRF Clean 3D will stop processing.

Once MRF Clean 3D starts processing, the “Status” field will display a series of dots and blank spaces (e.g., “..... !”). Each dot represents a completed process and each blank space represents a process to be completed. The “Command” field will display a percentage or a rotating bar indicating the current process is still running.

- (13) When completed, MRF Clean 3D will display a message that shows the time taken to process the specified input.

You may now open the output design file to see the results.

Batch Process

To run the batch process, simply key in the following in the Microstation command line:

```
Mdl load mrfcln3d -b[batch parameter file]
```

```
Ex. Mdl load mrfcln3d -bc:\batchprocess.txt
```

The batch parameter file should contain lines of text in the following format:

```
<input dgn><output dgn><info file><report file>
```

Ex.

```
<c:\input1.dgn><c:\output1.dgn><c:\input1.inf><c:\report1.txt>
```

```
<c:\input2.dgn><c:\output2.dgn><c:\input2.inf><c:\report2.txt>
```

```
<c:\input3.dgn><c:\output3.dgn><c:\input3.inf><c:\report3.txt>
```

You can also load Microstation and run the batch process automatically through DOS prompt or a Windows batch job with the following command line:

```
[Microstation command] [input dgn] -s[script file]
```

```
Ex. "ustation.exe" c:\input1.dgn -sc:\cleanactions.txt
```

In the script file (c:\cleanactions.txt), put the MRFCln3D batch command.

```
Ex. Mdl load mrfcln3d -bc:\batchprocess.txt
```

The batch process will run as soon as Microstation loads.

Advanced Topics

What to do when different features are on the same level

You can use the *edg* tool supplied with MicroStation to move the feature you want to clean to a unique level. After cleaning, you can move the feature back to the original level.

If you have MGE installed on your computer, you can create a ULF file containing the desired elements and use the ULF file as input to MRF Clean 3D.

Error messages

When running MRF Clean 3D interactively, error messages are displayed in the “Error” field of the MicroStation Command Window.

When running MRF Clean 3D in batch, error messages are written to the report file.

Memory requirements

MRF Clean 3D utilizes technologically advanced algorithms to account for all possible element positions, data structures and user selected options during processing. This requires an amount of memory equal to 100 times the input file. For example, if your input file is 0.5 MB, then you will need about 50 MB of memory to run MRF Clean 3D, which may consist of a combination of RAM and Virtual Memory. When you do fence cleaning, the amount of memory needed is equal to 100 times the size of elements within the fence. You can do the following to reduce the memory requirement when cleaning large files:

- Only clean related levels simultaneously. Unrelated levels can be cleaned in separate processes.
- Use fence cleaning option (**Use Current**, **Fence File** or **Auto Partition**) to clean a small part of the file each time.

The **Auto Partition** option is designed to clean a large DGN file (e.g. 10 MB) in one run. The amount of memory needed is equal to 100 times the partition size

The **Fence File** option is designed to clean the elements only within fences defined in the fence file.

You can add the fences to the **Out Fence File** and reference the fence file to avoid missing a certain area or processing an area multiple times. The fences should overlap a little to ensure that all elements are cleaned.

Efficiency

The efficiency of MRF Clean 3D depends on the total number of input points, the amount of RAM available, and the user-specified tolerances for the selected levels. If two levels are not related at all, they should

be cleaned separately by running MRF Clean 3D with the “By Level” toggle ON or by running MRF Clean 3D multiple times with separate levels selected each time.

Using the **Fence File** or **Auto Partition** option can dramatically reduce the time required to clean a large file. For example, to clean a 5.6 MB land use file took three hours using the **Auto Partition** option (partition size: 0.4 MB), whereas to clean the same file with the **No Fence** option took 12 hours.

When the number of input points is large, it may be more efficient to run MRF Clean 3D a few times using increased tolerances. The initial cleaning may reduce the size of the file by weeding redundant vertices and also reduce the number of elements by merging many shorter elements into one element. For example, if you want to use a tolerance of 50.0 meters for level 10 only, you can run MRF Clean 3D three times as shown below:

	Input	Output	Level	Tolerance (meter)
First time	raw.dgn	new1.dgn	10	10.0
Second time	new1.dgn	new2.dgn	10	25.0
Third time	new2.dgn	new3.dgn	10	50.0

Table 3: Processing a file several times

The output file NEW3.DGN is the final result. Intermediate files such as NEW1.DGN, NEW2.DGN can be deleted after the cleaning. The above process can be run in batch. Increasing the tolerances will increase the processing time because MRF Clean 3D will search a larger distance for other elements.



Note: Do not try to fix all dangles with MRF Clean 3D. This may not always be possible. Beyond a certain point, increasing tolerances can have a detrimental effect. Small polygons, for example, might be removed by the clustering of the boundary vertices.

Part of successful data cleaning is knowing when to stop processing with MRF Clean 3D and when to begin manual editing with MicroStation tools or MRF Flag Editor.

Trouble shooting

If you have difficulties running MRF Clean 3D, try the following:

- Use full path for file names
- Use smaller tolerances
- Use fences
- Make sure that you are using the correct version of MicroStation
- Increase your virtual memory settings and/or your RAM
- Contact your authorized MRF dealer for assistance

Operating Tips

MRF Clean 3D is designed to provide two major capabilities: flagging and fixing. If you want to flag sub-tolerance-elements, true intersections, and dangles, you should:

- Turn off fuzzy intersection
- Specify a valid **Free End Flag Level**
- Turn off true intersection
- Specify a valid intersection **Flag Level**
- Turn off **Del Sub-tol-ele**
- Specify a valid **Sub-tol-ele Flag Level**

To create true intersections, fix dangles, and delete sub-tolerance elements, you should:

- Turn on fuzzy intersection
- Turn on true intersection
- Turn on **Del Sub-tol-ele**

You can still specify a valid **Free End Flag Level** to flag the free ends of elements that can not be corrected by MRF Clean 3D, due to non-existence of entities within the specified tolerance. As a general rule, larger tolerances lead to fewer free ends (see “Note” under *Efficiency* topic above).

Flagging true intersections

If you want to flag all the true intersections, do not turn on “Fuzzy Intersection”. Fuzzy intersection may correct an overshoot and create a node at the true intersection as shown in **Figure 27**.

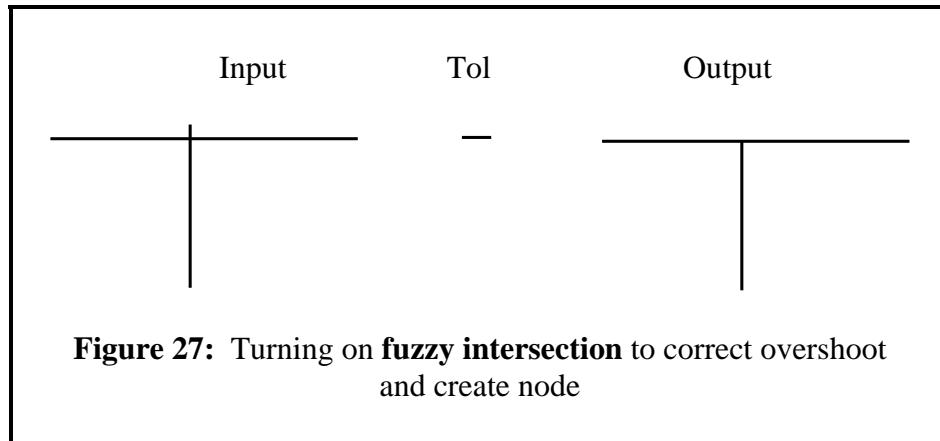


Figure 27: Turning on **fuzzy intersection** to correct overshoot and create node

Dangle removal

Turn on “True Intersection” if you want to remove more dangles. If “True Intersection” is not turned on, an overshoot (dangle) may have a length longer than (tolerance * **Dangle Factor**), because it is not broken at the true intersection.

Learning MRF Clean 3D with Example Design Files

You can use the supplied example design files to understand the cleaning effects of MRF Clean 3D. The MicroStation tool “Analyze” can be used to verify the output elements.

File name	Cleaning Goals	Levels	Tolerance (MU)
example1.dgn	Correcting overshoots undershoots	1	0.21
example2.dgn	Moving less accurate points to more accurate points.	1 2	0.21 0.01
example3.dgn	Processing line strings and removing dangles	1	0.06
example4.dgn	Processing parallel lines and removing duplicates	1	0.20
example5.dgn	Cleaning cells and lines	1 2	0.01 0.35
example6.dgn	Creating intersections	1	0.01
example7.dgn	Cleaning lines and cells with different settings of CELL_AS options. Test 1: set USE_CELL_AS: Node Test 2: set USE_CELL_AS: Non-node	1 2	0.01 0.40
example8.dgn	Cleaning lines with database linkages	1 2	0.21 0.01

Table 4: Example file descriptions

MRF Polygon *USER'S GUIDE*

For MicroStation V8/XM

May 2009



© MRF Geosystems Corporation
Suite 200, 625 – 14th Street N.W.
Calgary, Alberta
CANADA T2N 2A1

Tel: (403) 216-5515
Fax: (403) 216-5518
Email: mrfsales@mrf.com
WWW: <http://www.mrf.com>

Introduction

MRF Polygon (MRFPoly) is a polygon formation package. It takes a cleaned two-dimensional design file as input, and converts lines, line strings, and circular parametric arcs into complex shapes.

MRF Polygon can:

- Run on top of MicroStation either in batch or interactively
- Allow unlimited nesting levels of islands
- Group output polygons and their islands in one graphic group or group hole
- Handle circular parametric arcs without stroking
- Check one-to-one relationships between labels and formed polygons
- Polygonize only selected levels
- Flag error polygons for further editing

MRFPoly organizes the formed polygons so that no polygon hides another when it is filled. As a result, visualization and plotting can be readily done without any further preparation. Because error polygons are flagged, users can perform queued-editing of these errors using MRF Flag Editor (refer to *MRF Flag Editor User's Guide*). Polygons with no label or with more than one label will be set to the colors specified for easier identification and editing.

MRFPoly is ideal for cadastral mapping, municipal, and resource management applications.

POLYGONS
FORMED BY
MRF POLYGON

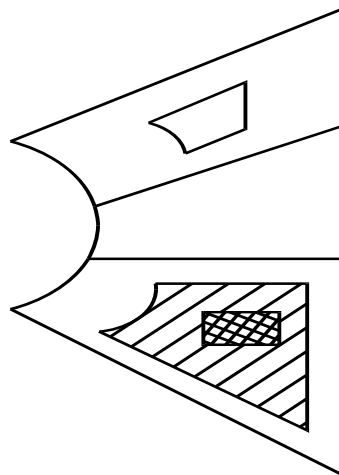


Figure 1: Example Polygons

Elements Processed by MRF Polygon

Type	MSTN Element Type/Number
Linework	LINE (3), LINE STRING (4), ARC ¹ (16),
Label (Area Centroid)	CELL (2), LINE ² (3), TEXT (17), TEXT NODE (7)

Note 1: Only circular arcs (where primary axis and secondary axis are of equal length) are processed by MRF Polygon.

Note 2: Only lines with zero length (i.e. the start point is the same as the end point) are considered as labels.

MRF Polygon Parameters

MRF Polygon parameters can be entered through a parameter file, which is optional for interactive processing, but required for batch processing. Parameters may also be entered interactively in the “MRF Polygon”, and “MRF Polygon Parameters” dialog boxes. This section will discuss how each parameter affects MRF Polygon processing.

MRF Polygon Parameters

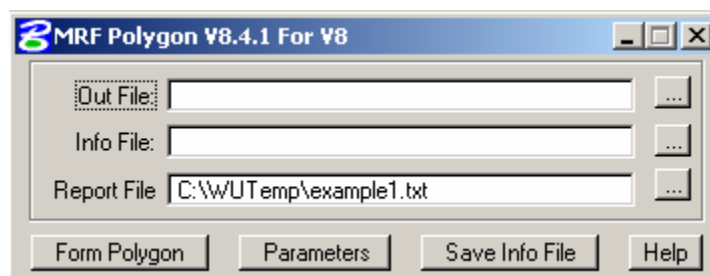


Figure 2: Main Dialog Box

Out File: (Optional, DGN file) This file holds the output. By default, the currently active design file is the output file (i.e. MRF Polygon will write the results back to the input file). You can activate the **file picker** (the push-button beside the “Out File” field) to key in an Out File.



Recommendation: It is recommended that you always specify an output file to keep the original design file intact. If you specify a design file which already exists, an alert dialog box will pop up and ask for permission to overwrite. In batch mode, for safety purposes, MRF Polygon will not overwrite an existing file.

Info File: (Optional, ASCII text file). This file contains the parameters for running MRF Polygon when you want to ensure the same set of parameters is applied to a set of files. Toggle button values are represented by 0 for OFF or 1 for ON. An option button can have as many values as the number of options: the first option is 1, the second 2, etc.

The INFO_FILE has the following format:

```
#This is the Info File (parameter file) for MRF Polygon
POLYGON_GROUP_OPTION:3
ISLAND_OPTION:1
LINE_WORK_OPTION:2
SYMBOLGY_INHERITANCE_LEVEL:1
OUTPUT_POLYGON_LEVEL:62
ERROR_FLAG_LEVEL:63
NO_LABEL_COLOR:0
MULTI_LABEL_COLOR:0
TOLERANCE:0.000001
SIMPLIFY_POLYGON:1
LEVEL_1:1
LEVEL_2:1
LEVEL_3:1
LEVEL_4:1
LEVEL_5:1
LEVEL_6:1
LEVEL_7:1
LEVEL_9:1
LEVEL_10:1
LEVELC_1:1
LEVELC_5:1
LEVELC_6:1
LEVELC_7:1
LEVELC_8:1
LEVELC_9:1
LEVELC_11:1
```

Note that while the format of the INFO_FILE is important, the ordering of the parameters is not. All uppercase characters before the colon (“:”) on each line in the above INFO_FILE template are considered to be keywords. There should be no space before a keyword. If a keyword appears in the INFO_FILE multiple times, the value will be set to the last appearance of the keyword. Therefore, MRF recommends using each keyword only once in the INFO_FILE to avoid confusion. Each parameter should occupy one line only. You can, however, add comments in the file as shown below:

```
POLYGON_GROUP_OPTION:2;this is to set a polygon group option
ISLAND_OPTION:2
LINE_WORK_OPTION:2
SYMBOLGY_INHERITANCE_LEVEL:5
```

```
The following line is to give a level for output polygons
OUTPUT_POLYGON_LEVEL:60
ERROR_FLAG_LEVEL:61
NO_LABEL_COLOR:0
MULTI_LABEL_COLOR:1
TOLERANCE: 0.001
```

```
LEVEL_1:1
LEVEL_2:1
LEVEL_3:1
LEVEL_4:1
LEVEL_5:1
LEVEL_6:1
LEVEL_7:1
LEVEL_9:1
LEVEL_10:1
LEVELC_1:1
LEVELC_5:1
LEVELC_6:1
LEVELC_7:1
LEVELC_8:1
LEVELC_9:1
LEVELC_11:1
```

In the above example, both “; *this is to set a polygon group option*” and “*The following line is to give a level for output polygons*” are comments added to the INFO_FILE. MRF Polygon will set parameters not specified in the INFO_FILE to defaults.



Note: You can activate the file picker, the push-button beside the INFO_FILE field, to select an INFO_FILE.



Note: Once an INFO_FILE is specified, MRF Polygon will load the parameters from the INFO_FILE and overwrite existing parameters.

Report File: (Optional, ASCII text file). MRF Polygon uses this file to log details of the MRF Polygon process and to echo the input parameters. The default file name is the first eight characters of the input DGN file name with a “.TXT” extension. For example, if your input file is C:\DGN\MRFTEST.DGN, then the default report file is C:\DGN\MRFTEST.TXT. If no report file is specified, MRF Polygon uses MRFPOLY.TXT in the current directory as the default. If the report file does not exist, MRF Polygon will create the report file. If the designated file already exists, MRF Polygon will append the following information to it:

- ◆ Input file name
- ◆ Input file size in bytes

- ◆ Input parameters
- ◆ Major MRF Polygon processes
- ◆ Number of polygons generated
- ◆ Number of line strings not formed into polygons
- ◆ Output file name
- ◆ Output file size in bytes
- ◆ Completion or ERROR message
- ◆ Elapsed time of the operation

You can read the report file periodically to monitor MRF Polygon's progress when MRF Polygon is running in batch under WINDOWS NT/95. The report file also helps you find the places where MRF Polygon encountered problems.



Note: The Report File is a prime source of information if you ever have a problem forming linework into polygons.

Help: When this button is pushed, an on-line User's Manual will be displayed. You can scroll through it for information about setting parameters.

Save Info File: When you click this button, MRF Polygon will pop up a file creation dialog box and ask you to key in a file name. All the current parameter settings will be saved to this file.

Parameters: When this button is pushed, MRF Polygon's Parameters sub-dialog box will open. Parameter values are read in from the INFO_FILE (if designated) or set to MRF Polygon default values if no INFO_FILE is specified. MRF Polygon will scan the input design file and disable the level check-boxes corresponding to the levels that have nothing to be processed.

You can accept the default values or make changes in the parameter sub-dialog box as desired.



Note: Once the parameter sub-dialog box is opened, the parameters in this sub-dialog box will be used in processing.

Form Polygon: When this button is pushed, MRF Polygon will start processing. If you want to abort MRF Polygon while it is processing, enter CTRL+C (“^C”). When aborting MRF Polygon under WINDOWS NT or WINDOWS 95, an “Alert” window will open, and you should click the “OK” button. Next, a “MicroStation” text window will open. If you do not want to exit MicroStation, do not close this text window.

MRF Polygon Parameters Input

Figure 3-A: MRF Polygon Parameters Dialog Box

Id	Level Number	Level Name	Select
96	218	Map Control Pts	FALSE
97	251	Lot-PCL-Blk Lin	TRUE
98	235	TR-PCL Map Line	TRUE
99	249	VAC Street Line	FALSE
100	201	City Bdry Line	TRUE
101	253	Cut Line	TRUE
102	211	Centerline Reco	FALSE
103	231	Private Street	TRUE

Figure 3-B: MRF Polygon Select Level Dialog Box

Polygon Group: (Default is “Graphic Group”) There are three options for grouping polygons: none, graphic group and group hole. If you choose “None” (option 1), the formed polygons will not be grouped. If you choose “Graphic Group” (option 2), the holes will be put into the same graphic group with their parent. Lastly, if you choose “Group Hole” (option 3), the holes and their parent will be used to create a group hole.

Island Option: (Default is “Keep All”) “Keep All” (option 1) will keep both the island polygons and holes. For example, a lake with an island will be represented as an outside polygon (lake boundary) minus the island polygon, and both the lake and the island will be output. “Delete Island” (option 2) will delete the island polygons. “Delete Holes” (option 3) will delete all the hole polygons.

Line Work Option: (Default is “Delete Existing Linework”). “Keep Linework” (option 1) will output the linework in the input file to the output file. If no output file is specified, the linework in the input file is kept as is. Under option “Delete Existing Linework” (option 2), the original linework, which has been used to form polygons, will not be written to the output file.



Note: If no output file is specified, the default “Delete Existing Linework” option will delete all linework which has been used to form polygons in the input file.

Symb. Inherit. Level: (Default is level 1) An output polygon may have elements from a few selected levels. If the polygon has at least one element (A) from the symbology inheritance level, the symbology of the output polygon will be set to A’s symbology. If an output polygon does not have an element from the symbology inheritance level, the output polygon’s symbology will be arbitrarily chosen from one of its elements.

Output Poly Level: (Default is level 62) Formed polygons will be output to this level.

Error Flag Level: (Default is level 63) If you specify a valid level in “ERROR_FLAG_LEVEL” field, MRF Polygon will place a text element ‘D’ at each error polygon element on that level to help you identify potential errors. Another MRF product, MRF Flag Editor, can be used to conveniently loop through the error polygon elements.

No Label Color: Cells, zero length lines, text and text nodes on the selected levels will be considered to be labels for polygons. This option allows you to specify

the color for polygons with no labels inside them. MRF Polygon will then mark all no-label polygons with this color. A label is considered inside a polygon if its position is inside the polygon. The position of a text element is its snap point, and the position of a text node element is its user-defined origin.

Multi Label Color: Polygons that have more than one label inside them will be area-filled with this color. If more than one point, cell, text or text node have positions inside a polygon, the polygon is considered to have multiple labels.

Tolerance: (Default is 0.000001, in Master units) Arc end points may not be snapped to the end points of line strings or arcs—even after cleaning—because of the way MicroStation represents and displays arc elements. If you use arcs to form polygons, you must specify a tolerance (in Master units). MRF Polygon will search for an arc or line string within the tolerance of an arc end when forming polygons.

Special Case Hole: If this toggle button is checked special holes will be formed. A special hole is a hole shares a point with its parent polygon.

In Figure 4, when special case hole is On, MRF Polygon will treat polygon B as a hole of Polygon A. When special case hole is Off, MRF Polygon will create two positive polygons: A and B.

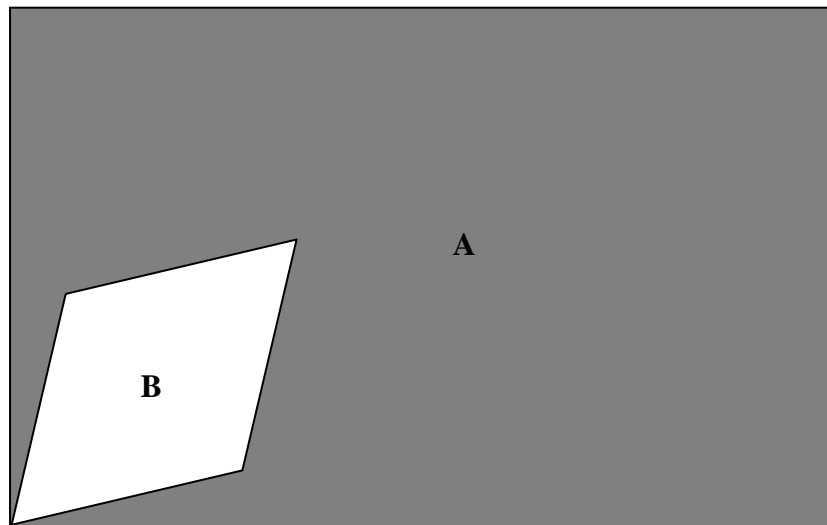




Figure 4: An Example of a Special Hole

Input Linework Levels: This text box allows user to key in the line work levels.
Ex. To process level 1 and level 3, simply key in “1,3” in the text box. Users

can also click the  button on the right side to start the Select Level Dialog Box (Figure 3-B) to select the input linework levels interactively. To select a level, simply click a level and set the “select” to TRUE.

Select label Levels: This text box allows user to key in the Label levels. Users can also click the  button on the right side to start the Select Level Dialog Box to select the input label levels interactively. To select a level, simply click a level and set the “select” to TRUE.

Running MRF Polygon

Follow these steps to load and run MRF Polygon interactively:

- (1) Start MicroStation and open the design file to be polygonized.
- (2) Choose “MDL Applications” from the “User” pull-down menu.
- (3) Click at “mrfpoly.ma” using the mouse.
- (4) Click push-button “Load”.

MRFPoly is now loaded and its main dialog box (see **Figure 2**) will appear.



Tip: Steps (2) - (4) can be replaced by issuing one command:

mdl load mrfpoly

under MicroStation's command window.

- (5) Specify the full path for the output design file in the OUT_FILE text field. If the specified file already exists, MRF Polygon will issue a warning and ask you for permission to overwrite. If this field is not specified, the results will be written to the current design file. If you do not specify the full path, the file will be created in the current directory.



Note: You can activate the file picker beside the OUT FILE field to select an existing directory.

- (6) Optional: Specify the INFO_FILE that contains the parameters for running MRF Polygon. Alternatively, you can activate the file picker beside the INFO FILE field to select an INFO_FILE. If you have such a file, you may go directly to step 11 to run MRF Polygon, or you may go to the next step to modify the parameters read from the INFO_FILE.
- (7) Press “Parameters” to bring up the parameters sub-dialog (see **Figure 3**).
- (8) There are several things you can set in this box:

- a) Choose a Polygon Group Option. This option is initialized to "Graphic Group", (i.e. a polygon and its holes are grouped together with the same graphic group number).
 - b) Choose an Island Option.
 - c) Choose a Linework Option.
 - d) Set the Error Flag Level by entering a level number in the text field. A "D" text element will be placed on this level to mark the location of each error detected.
 - e) Set "Output Poly Level" and "Symb Inheritance Level".
 - f) Set "No Label Color". If a polygon has no label, it will be area-filled with the "No Label Color".
 - g) Set "Multi Label Color".
 - h) Enter a proper tolerance in Master units, if there are arcs in the design file.
- (9) Check the "Special Hole", if you want to form special holes.
 - (10) Select Line work levels on the Level Map of "Select Linework Levels:"
 - (11) Select Label levels on the Level Map of "Select Label Levels:"
 - (12) Click on "MRFPolygon" to start the polygon formation process. You may close any opened sub-dialogs before doing so.



Note: If on or more input parameters are incorrect, MRF Polygon will stop processing and display an error message in the "Error" field of the MicroStation Command Window.



Note: If you wish to stop MRF Polygon in the middle of processing, press CTRL+C.

- (13) Open the output DGN file to see the result.



Note: Click on "Help" at any time to get *on-line* help.

Batch Process

To run the batch process, simply key in the following in the Microstation command line:

```
Mdl load mrfpoly -b[batch parameter file]
```

```
Ex. Mdl load mrfpoly -bc:\batchprocess.txt
```

The batch parameter file should contain lines of text in the following format:

```
<input dgn><output dgn><info file><report file>
```

Ex.

```
<c:\input1.dgn><c:\output1.dgn><c:\input1.inf><c:\report1.txt>
```

```
<c:\input2.dgn><c:\output2.dgn><c:\input2.inf><c:\report2.txt>
```

```
<c:\input3.dgn><c:\output3.dgn><c:\input3.inf><c:\report3.txt>
```

You can also load Microstation and run the batch process automatically through DOS prompt or a Windows batch job with the following command line:

```
[Microstation command] [input dgn] -s[script file]
```

```
Ex. "ustation.exe" c:\input1.dgn -sc:\cleanactions.txt
```

In the script file (c:\cleanactions.txt), put the Mrfpoly batch command.

```
Ex. Mdl load mrfpoly -bc:\batchprocess.txt
```

The batch process will run as soon as Microstation loads.

Recommended MRF Polygon Work Flow

Before running MRF Polygon, you must make sure that the input design file is clean and has:

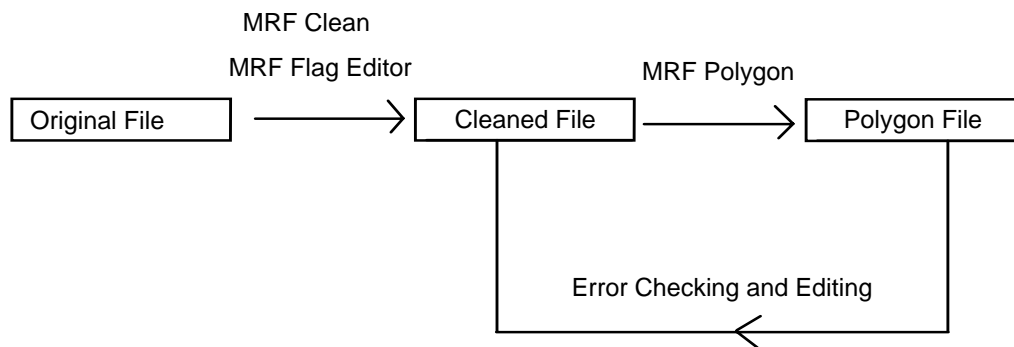
- no duplicate line work
- no overshoots or undershoots
- linework broken at all intersections

You can use **MRF Clean** and **MRF Flag Editor** to clean your input files before running MRF Polygon.

Even if your input file has been cleaned, MRF Polygon may find additional linework problems. You will need to fix these for the process to run successfully.

MRF recommends that you use one of the two workflows shown below. You are encouraged to experiment with both to find the one best suited to your environment.

Work Flow 1:



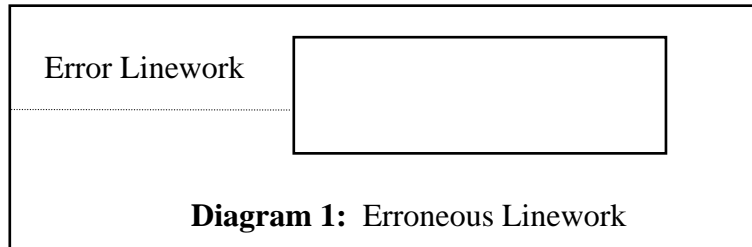
1. Run MRF Clean on the original file (*original.dgn*) to get an output file (*clean.dgn*). The following are the key parameter settings when running MRF Clean:

Remove_Duplicate: By Geometry
By level: Off
Fast: Off
Fuzzy_Intersection: On
True_Intersection: On
Del_Sub_Tol_Ele: On
Free_End_Flag_Level: 63 (or another empty level)
Tolerance: All the levels used to form polygons should have valid tolerance.

Other parameters are not crucial.

2. Examine the output file *clean.dgn*. You can use MRF Flag Editor to edit the “D” flags and delete danglers. Repeat steps 1 and 2 if necessary until a clean file is produced (a file without “D” flags).

3. Run MRF Polygon on *clean.dgn* with the “Delete existing linework” option, to obtain the output file *poly.dgn*.
4. Examine *poly.dgn*. All the linework which has not been used to form polygons, remains on its original levels in *poly.dgn*. Check for any erroneous linework like the dashed line in Diagram 1.

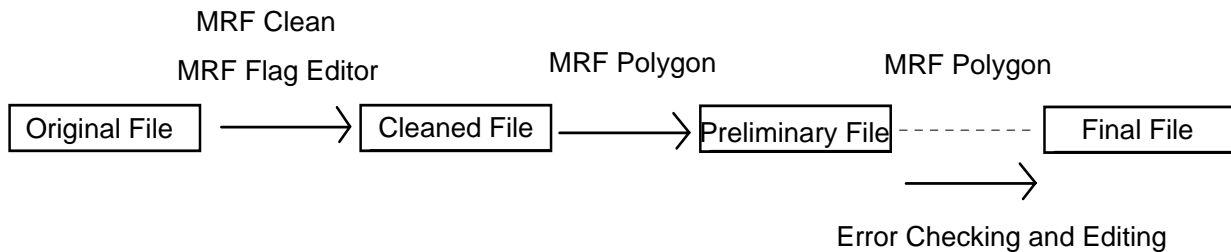


To find the erroneous linework as shown in the above diagram:

- a) turn on the original linework levels
 - b) zoom into one original linework
 - c) turn on the “Output_Polygon_Level”
 - d) turn on fill (View → Attribute → Fill)
 - e) check if the original linework is erroneous
5. If *poly.dgn* contains erroneous linework as shown in the above diagram, do the following to delete or edit each instance of erroneous linework:
 - a) zoom into the erroneous linework in *poly.dgn*
 - b) key-in “*xd=clean.dgn*” in MicroStation Key-in field. MicroStation will open *clean.dgn* with the same view extent.
 - c) delete the erroneous linework or edit it.
 - d) key-in “*xd=poly.dgn*” to return to *poly.dgn*.

If you have edited the linework, you may need to run MRF Clean again on *clean.dgn*.

6. Run MRF Polygon again using the edited *clean.dgn* as input.
7. Repeat steps 4 to 6 if necessary until all the polygons are formed.

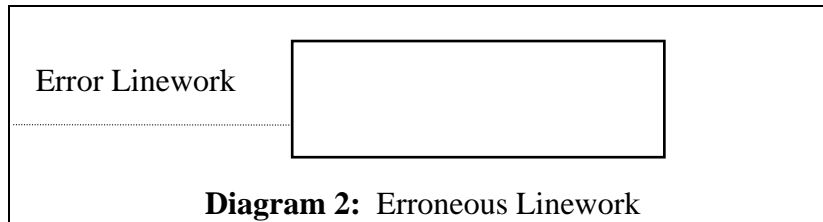
Work Flow 2:

1. Run MRF Clean on the original file (*original.dgn*) to get an output file (*clean.dgn*). The following are the key parameter settings when running MRF Clean:

Remove_Duplicate: By Geometry
By level: Off
Fast: Off
Fuzzy_Intersection: On
True_Intersection: On
Del_Sub_Tol_Ele: On
Free_End_Flag_Level: 63 (or another empty level)
Tolerance: All the levels used to form polygons should have valid tolerances.

Other parameters are not crucial.

2. Examine the output file *clean.dgn*. You can use MRF Flag Editor to edit the “D” flags and delete dangles. Repeat steps 1 and 2 if necessary until a clean file is produced (i.e. a file without “D” flags).
3. Run MRF Polygon on *clean.dgn* with “Keep linework” option, to obtain a preliminary output file *poly.dgn*.
4. Examine *poly.dgn*. All the input linework remains on their original levels in this file. If you find any area where polygons are not formed as expected, check whether erroneous linework exists—like the dashed line in Diagram 2.



To find the erroneous linework as shown in Diagram 2:

- a) turn on the original linework levels
 - b) turn on the output polygon level
 - c) turn on the output polygon level
 - d) turn on fill (View → Attribute → Fill)
 - e) visually locate the erroneous linework
5. If *poly.dgn* contains erroneous linework as shown in the above diagram, delete the erroneous linework or edit it. If you edit the linework, you may need to run MRF Clean again on *poly.dgn*.
 6. Delete all the previously formed polygons on the output polygon level in *poly.dgn*.
 7. Run MRFPoly again using the edited *poly.dgn* as input. MRF recommends that you use a different output file name (e.g. *poly1.dgn*).
 8. Repeat steps 4 to 7, if necessary, until all the polygons are formed.

Learning MRF Polygon with Example Design Files

You are encouraged to use the supplied example design files to discover how MRF Polygon works. The MicroStation tool “Analyze” can be used to verify the output elements.

EXAMPLE1.DGN: Form polygons

Linework Level: 1
Label Level: 1

EXAMPLE2.DGN: Island handling

Linework Level: 1
Polygon Group: Graphic Group
Island Option: Keep Island

Turn on the “Graphic Group” setting. Move the formed polygons and examine them.

EXAMPLE3.DGN: Island handling

Linework Level: 1
Polygon Group: None
Island Option: Delete Holes

EXAMPLE4.DGN: Form polygons from elements on different levels

Linework Level: 1, 3
Label Level: 1, 3
Polygon Group: None
Island Option: Delete Holes
Symb. Inheritance Level: 1

Polygon 301 has the same symbology as polygon 101.

EXAMPLE5.DGN: Error polygon and polygon labels detection

Linework Level: 1
No Label Color: 1
Multi Label Color: 5

This example DGN file will have an error polygon and the character “D” will be placed to flag the error element. The symbology and size of the character “D” was taken from the active settings. If you do not like the existing active settings, you can change these settings before running MRF Polygon. Color one will be assigned to the formed polygons without any label. The formed

polygons with only one label within will have a color of their own, and the formed polygons with more than one label will have a color of five.

EXAMPLE6.DGN: Arc involved in polygon formation

Linework Level: 1, 3, 21, 63
Tolerance (MU): 0.01