

1125 16th Street, Suite 213
Arcata, CA 95521
(707)-822-8005

pt_mend Documentation
version 3.0
April 1, 1999

- Installation Instructions -
 - Purpose -
- Execution and Options -
 - Examples -
 - Hints -
 - Support -

README
pt_mend Documentation

Installation Instructions:

UNIX: pt_mend is delivered on a DOS formatted floppy. Set your working directory to the destination directory (e.g. /usr/bin) in which you want to place pt_mend and its associated processes.

Copy the file 'grs_pt' off the floppy by keying in the following command:

```
dtu grs_pt grs_pt
```

Then make the file 'grs_pt' executable by keying in:

```
chmod 777 grs_pt
```

Then key in:

```
grs_pt
```

This initiates a process that will copy all the necessary files from the floppy to the

destination directory and make them executable.

Windows NT: pt_mend is delivered on a DOS formatted floppy or as a self extracting pkzip exe file. Set your working directory to the destination directory (e.g. c:\bin) in which you want to place segjoin and its associated processes.

Copy the file 'setup_pt.exe' off the floppy using the File Manager, Explorer, or Then execute this file by selecting it.

This initiates a process that will unzip all the necessary files from the container file to the destination directory.

UNIX and Window NT: As long as the destination directory is in your PATH, you can run segjoin from any directory. If your destination directory is not in your PATH, modify your PATH and add the destination directory to your PATH statement. Your destination directory should not be a system directory that may be deleted during either a product or system software removal or delivery.

pt_mend support:

pt_mend is developed, maintained, and supported by Geographic Resource Solutions of Arcata, CA. Should you need support or information, or if you have suggestions or feedback, please contact:

Kenneth A. Stumpf
Geographic Resource Solutions
1125 16th Street, Suite 213
Arcata, CA 95521

(707)-822-8005
(707)-822-2864(fax)

*****.*

Program Purpose:

*****.*

pt_mend is a process that is useful in the cleaning of line work and points that will eventually be used to form areas or linear networks. The main purpose of pt_mend is the identification of X,Y coordinates that fall within a user specified tolerance or distance and the modification of those points to a common coordinate. If arcs are to be included in the processing they must first be stroked to a line string using a process such as pt_mend or linecleaner. Modified linear elements can be separated from the original linework and the original elements can be retained or deleted. Output symbology may be specified, as well as the element type (lines and line strings versus curves). Questionable endpoints may be identified and flags placed. The process only considers end points of lines, line strings, arcs, cells, text, text nodes and curves.

All X,Y coordinates may be processed by first running segjoin and decomposing the linework into lines with only two vertices, then running pt_mend to modify the coordinates, and then running segjoin to put the linework back together.

Execution: pt_mend is initiated differently in the UNIX and Windows/WinNT environments, as follows:

UNIX:

To run the pt_mend process from the command line key-in :

```
pt_mend [args]
```

Windows/WinNt:

To start the GUI from the command line key-in :

```
pt_mendf
```

To run the pt_mend process from the command line using perl key-in :

```
perl -S pt_mend.pl [args]
```

There are a number of options [args] that may be specified when you run pt_mend. You only need to include on the command line the options you desire, otherwise process defaults will be used.

```
-c
-d
-e [ input_element_type(s) ]
-f [ flag_level,flagSize ]
-fP [ flag_float_level,flagSize ]
-lc [ input_class(es) ]
-ll [ input_level(s) ]
-lo [ input_color(s) ]
-ls [ input_style(s) ]
-lw [ input_weight(s) ]
-s
-w[x] [ value,weight ]
-z
-AS
-D [ dflag_level,dflag_size ]
-F [ dgn_filename ]
-FR [ adj_dgn_filename ]
-L [ output_level ]
-LM [ multiple_line_level,multiple_line_flag_size ]
-LMM [ multiple_line_level ]
-LS [ single_line_level,single_line_flag_size ]
-LSM [ single_line_level ]
-NL
-NP [ pointLimit ]
-NS
-O [ IntValue, ModValue, AveValue ]
-Pf [ flagLevel,flagSize ]
-PF [ flagLevel,flagSize ]
-PM [ flagLevel,flagSize ]
-PMM [ multiple_point_level ]
-PS [ single_point_flagLevel,single_point_flagSize ]
-PSM [ single_point_level ]
-S [ style,weight,color,class ]
-T [ tolerance(distance) ]
-U [ ulfname ]
-XL
```

subject to the following definitions:

```
-c signifies to output curves rather than linestrings.
```

- d signifies to output diagnostic messages.
- f [flagLevel,flagSize] signifies to identify questionable points and place flags of size flagSize on the level indicated by the value of flagLevel. No elements are modified when this flag is selected.
- fP [flag_float_level,flag_size] signifies to identify floating points and place flags of size flag_size on the level indicated by the value of flag_float_level. No elements are modified when this flag is selected.

Design file input criteria:

- e [input_element_types] specify the types to process; the default values are types 3 and 4.
- ll [input_levels] specify the levels to process; the default values are 1 - 63.
- lo [input_colors] specify the colors to process; the default values are 0 - 255.
- ls [input_styles] specify the styles to process; the default values are 0 - 7.
- lw [input_weights] specify the weights to process; the default values are 0 - 30.

Input criteria are comma separated integer values. Ranges may be signified with a dash ('-').

For example: -ll 1,2,3,11-20

- s signifies to save the original linework - the default mode is to delete the original linework. Original linework will only be saved when this flag is included and the mended linework is output to another level as specified with the -L flag.
- w[x] [indexValue,weight] signifies to uses priorities in determining solutions using weights as indicated by the arguments x, indexValue, and weight. Argument x indicates to use weights based on either element level (l), color (o), or type (t). Up to ten sets of -w flags may be included on the command line. Weights can range from 0-255. For level weighting, the indexValue indicates the level number. For type weighting, the indexValue indicates the element type. For color weighting, the indexValue indicates the element color(0-255). Higher weight values indicate higher levels of priority in solving conflicting points.
- z signifies to reset Z values to zero(0).
- AS signifies to strip attribute linkages from output /modified elements.
- D [dflag_level,dflag_size] signifies to flag dangling end points and place flags of size = dflag_size on dflag_level.
- F [dgn_filename] specifies the name of the design file (dgn_filename) to process. THIS OPTION SHOULD NOT BE USED IF DATA IS BEING SELECTED BY SPECIFYING A .ulf FILE WITH THE -U OPTION. All linear elements in the file will be processed unless selection criteria are specified.
- FR [adj_dgn_filename] specifies the name of an adjacent design file (adj_dgn_filename) to process. THIS OPTION SHOULD NOT BE USED IF DATA IS BEING SELECTED BY SPECIFYING A .ulf FILE WITH THE -U OPTION. All linear elements in the file will be processed unless selection

criteria are specified. Up to ten adjacent files may be processed, with each `adj_dgn_filename` preceded by `-FR`.

- L [`out_level`] signifies to output modified linework to the level indicated by `out_level`. Valid level values range between 1 and 63. The default is the original level of the element. If linework is output back to its original level, the original linework will be deleted (the `-s` option will be overridden). Cells, text, and text nodes will stay on their original levels.
- LM [`multiple_line_level,multiple_line_size`] signifies to identify multiple linear elements by placing flags of size `multiple_line_size` on `multiple_line_level`.
- LMM [`multiple_line_level`] signifies to move multiple linear elements to `multiple_line_level`. The first linear element of the group will remain on its original level.
- Nm signifies to use the average x,y values to resolve points rather than use a modal solution, if one is possible. The intersection solution is the default solution, if one exists. If one does not exist, the modal solution methodology is used, if a modal solution exists. The average approach is used when the intersection and modal solutions are not possible (they do not exist).
- NL signifies not to check for 'linked' elements. Elements are 'linked' if they have a common X,Y coordinate at one end of each linear element. Ignoring 'linked' elements may result in collapsing different linear elements into a common line. The default mode is to check for 'linked' elements, and if they are found, do not mend the other endpoints of these elements.
- No signifies NOT to check for closed loops when using the `-LM` flag.
- NP [`pointLimit`] signifies NOT to consider groups of points that have at least `pointLimit` points in the group.
- NS signifies NOT to check for elements with both endpoints within the specified tolerance.
- O [`IntValue, ModValue, AveValue`] - these arguments enable the specification of priorities for the different mending approaches - by intersection, by modal solution, and by averaging X,Y point values. Values (priorities) must be between 0 and 3 - with a higher value assigning precedence in the solution order.
- Pf [`flagLevel,flagSize`] signifies to flag FLOATING point locations, placing flags of size `flagSize` on `flagLevel`.
- PF [`flagLevel,flagSize`] signifies to flag MENDED point locations, placing flags of size `flagSize` on `flagLevel`.
- PM [`flagLevel,flagSize`] signifies to flag MULTIPLE point locations, placing flags of size `flagSize` on `flagLevel`.
- PMM [`multiple_point_level`] signifies to move MULTIPLE point features to `multiple_point_level`. The initial point will be left on its original level.
- S [`style,weight,color,class`] signifies to output linework using the element symbology characteristics that follow in the command string. `style,weight,color,class` are integer values that represent the element output symbology. Input items are order dependent and comma separated.

- T [tolerance(distance)] signifies the magnitude of the tolerance within which points will be mended. This distance is expressed in terms of design file master units (MUs) and may be expressed as a decimal portion of MUs.
- XL signifies to ignore differences in levels when determining whether or not duplicate linear elements are present.
- U [ulfname] specifies the ulf filename in the command string and the name of the ulf file (ulfname) that identifies the linework to be processed. This file may be created from within MicroStation by identifying an area with a fence and keying in "el=filename.ulf" or, if you have MGE, by running ulfbuilder. If neither the '-U ulfname' or the '-F dgnname' are specified on the command line the process will invoke ulfbldr to generate a ulf file. THIS OPTION SHOULD NOT BE SELECTED IF DATA IS BEING READ DIRECTLY FROM A DESIGN FILE USING THE -F OPTION.

pt_mend hints:

We recommend that design files be run through linecleaner, to remove duplicate linework, both before and after running pt_mend. Duplicates may be created after modifying end points. We advise, as a precaution, that you backup your files before processing. We have not lost any files, but that does not mean the process is infallible.

Input can be from either a design file or a ulf file. Design file input requires the specification of element types, levels, and symbolic properties (such as style, weight, color, and class) if these characteristics are to be used to limit the number of elements processed. These specifications are included as command line input to pt_mend. ulf files may be created before running pt_mend using a fence and "el=ulfname" or by running ulfbuilder (an MGE process). The ulfname may then be included after the -U flag on the pt_mend command line. If a ulf file is not specified on the command line and you have MGE, the process will initiate ulfbuilderA or ulfbuilderF. These processes can only be run after initializing mge by keying in 'mge -u -p projectname'. You can select whichever version of ulfbuilder you wish to be invoked by editing the pt_mend shell script and uncommenting/commenting the appropriate line. When either ulfbuilderA or ulfbuilderF are invoked, you must Execute and Continue out of these windows. Following ulfbuilder... a message 'Building ulfname' will be printed to your terminal. If an error condition is detected, (e.g. no ulfname) the process will prompt you to re-enter the ulf being run locally, but is accessing files off a server or another node on the network. If pt_mend hangs up, the solution is to abort the process, run ulfbuilder... separately, and then run pt_mend and specify the -U ulfname option on the command line.

In the event GRS_pt_mend abnormally aborts after opening a design file, or if you interrupt a run, the input design file may be locked from further write access. This situation may be resolved by killing the ustation process associated with GRS_pt_mend that is still running on the system. It is alright to abort the process up until it is writing elements back into the file. If

the process is aborted during this stage some data (a line or two) may be lost.

Temporary workfiles, `dimen.$$`, `startstp.$$`, and `sort_pts.$$`, are written to the present working directory. These filenames should not be in conflict with existing files, or they will be overwritten. These files are deleted automatically following the successful execution of the process. In the event that the process aborts abnormally these work files may need to be removed by the user.

`pt_mend` will process up to 250,000 lines per run. If more lines exist, continue to resubmit the process, outputting the processed linework to a new level, until the file is completed. If the file is very large, it may be necessary to identify portions of the file using a fence (`el=ulfname`) to break the file into smaller processing units.

*****.*

`pt_mend` examples:

*****.*

from the command line:

```
pt_mend -U dummy.ulf -T 5 -Nm
```

```
pt_mend -U /usr/mge/%grs/ulf/lev60.ulf -L 40 -s -T 0.5
```

```
pt_mend -F cover.dgn -L 10 -S 1,,5, -T 1 -Nm
```

```
pt_mend -F pix_lines.dgn -L 2 -S 0,1,4 -T 25 -ll 1,4 -ls 0
```

```
pt_mend -U cover -ll 1-4 -lo 4 -lw 0 -f 63,100
```

for a shell script application:

```
for i in `find ./ -name "*.dgn" -print`  
do  
  pt_mend -F $i -T 2 -L 5  
done
```

*****.*

Support:

*****.*

`pt_mend` is developed, maintained, and supported by Geographic Resource Solutions of Arcata, CA. Should you need support or further information, or if you have suggestions or feedback, please contact:

Kenneth A. Stumpf
Geographic Resource Solutions
1125 16th Street, Suite 213
Arcata, CA 95521

(707)-822-8005
(707)-822-2864(fax)