

1125 16th Street, Suite 213
Arcata, CA 95521
(707)-822-8005
(707)-822-2864 (fax)
email: grs@northcoast.com

segjoin Documentation
version 5.6
September 12, 2001

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

README
segjoin Documentation

```

/*****/
Installation Instructions:
/*****/
```

UNIX: segjoin 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 segjoin and its associated processes.

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

```
dtu grs_sj grs_sj
```

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

```
chmod 777 grs_sj
```

Then key in:

```
grs_sj
```

This initiates a process that will copy all the necessary files from the floppy to the destination directory and make them executable.

Windows/WinNT: segjoin is delivered on a DOS formatted floppy or by email in a

pkzip exe file. Set your working directory to the destination directory (e.g. c:\grs) in which you want to place segjoin and its associated processes. Place the file 'setup_sj.exe' in this directory and then execute this file using the appropriate tools.

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.

segjoin support:

segjoin 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
Voice: (707)-822-8005 FAX:(707)-822-2864 email : grs@northcoast.com

/*
Program Purpose:
*/

segjoin is a linear data processor that was originally developed to join individual lines and linestrings into longer multi-vertice linestrings.

segjoin is designed to remove unnecessary breaks in lines, linestrings, curves, connected strings, arcs, and b-spline curves. Arcs, b-splines, and shapes can be converted to linear elements as part of the process. Linear elements may be strung together removing unnecessary breaks that result in extra element headers and redundant starting and ending vertices. The default mode of the process is that linework of different levels, symbolic properties, graphic groups, or zlow/zhigh range values will not be joined. Specific options enable the user to override these defaults. Linework may be processed level by level or for a group of levels at one time. Original linear elements may be saved (-s flag) if the joined linework is placed on another level. When multiple levels are processed all intersections of linear elements from all levels are used to determine where a linear element ends. Cells, text, and text nodes can be included during processing to retain breaks at these points. Dangling endpoints can be identified using the -D flag and linework can include data in adjoining reference files tocheck connectivity between elements of adjacent files.

As segjoin evolved over the past seven years it has been enhanced to include many different types of linear functionality. While the major purpose of segjoin is element consolidation, the process also performs other significant functions; these functions include line weeding, resymbolization, element conversion, label creation, database loading, element deletion, and cleaning. Linework can be weeded using the -W flag. Weeding options include distance, length, and angle tolerances. Linework may be output with user specified symbolic properties of style, weight, and color, so segjoin can be used to resymbolize data. The "class" of the linework may also be altered by this process by specifying a different class as one of the options following the -S flag. Linework may be output to a new level using the -L flag, or as one of several element types, including lines, linestrings, curves or b-splines, or as connected (complex) strings (if none of these element properties are specified the process outputs the elements as linestrings and lines using their

original symbolic specifications). Labels for linear elements can be created from the database and placed along the linear elements (e.g. street names on street centerlines).

segjoin may also be used to manipulate "zero-length" lines by either deleting or moving them to another level. Short length segments can be removed using the -O option. Dangling end-points may be identified using the -D flag. segjoin will also remove consecutive identical vertices found in the definition of a linear element. Elements may be deleted using the -delete flag. Linear values that include slope, distance, and from/to x,y, and z values can be loaded into the database using the -XY option. Attribute linkages can be manipulated, including their removal using the -A flag.

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

UNIX:

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

```
segjoin [args]
```

Windows/WinNt:

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

```
segjoinf
```

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

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

There are a number of options [args] that may be specified when you run segjoin. You only need to input (on the command line) the options you desire, otherwise process defaults will be used. segjoin is run by keying in the command line:

```
[-U ulf_filename ]      [-sa arc_str_tol]      [-sc curve_str_tol]
[ [-F dgn_filename]    [-FR ref_filename]
  [-E ent_num(s)]      [-u user_ent_word]
  [-e input_element_type(s)]  [-ll input_level(s)]
  [-lo input_color(s)]  [-lw input_weight(s)]
  [-lc input_class(es)] [-ls input_style(s) ]
[-L outlevel [,llev,lsize] ]  [-D dlev,dsize]  [-R]
[-S s,w,co,cl [,slev,ssize] ]  [-s]  [-V num_verts]
[-WR pixel_size] [-Z zero_in]  [-z]  [-C]  [-b]  [-c]  [-Ax ]
[-W d_tol,l_tol,a_tol]  [-WZ]  [-Nx [] ]  [-f]  [-g]  [-delete]
[-label
```

```
table,column,level,size,font,justif,color,weight,angle,minDist,labSpacing,xoffset,yoffset]
[-XY      tablename,slopeColumnName,slopeOpt,lengthColumnName,lengthOpt,
          FromXCol,FromYCol,FromZCol,XYZOpt,ToXCol,ToYCol,ToZCol ]
[-O[F]  dist[,level,size]]  [-SH centroidType,level[,font,justification,size]]
```

subject to the following definitions:

Input Options:

```
[-F dgn_filename]:
```

The -F flag precedes 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. Dgnfile input selection criteria are as follows:

[-E ent_num(s)]: default is all. Enter the list of entity numbers to selectively process.

[-u user_ent_word]: default is 0. Enter the word offset of the entity number in the user data linkage, if user data linkages are to be processed.

[-e input_element_types]: default is types 3 and 4.

[-ll input_levels]: default is 1-63.

[-ls input_styles]: default is 0-7.

[-lw input_weights]: default is 0-30.

[-lo input_colors]: default is 0-255.

[-lc input_classes]: default is 0,2.

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

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

Note: Curves and arcs may be included for processing by including them as element types to extract (-e 3,4,11,16). Curves will not be stroked unless the -sc flag is specified. Arcs will be stroked using the system default if the -sa flag is included.

[-FR ref_filename]:

The -FR flag precedes the name of a reference design file (ref_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 using selection criteria as specified. Up to ten reference files can be included, each with a separate -FR flag.

[-sa [arc_str_tol]]:

The -sa flag signifies to include arcs and stroke them using the specified tolerance arc_str_tol. The tolerance indicates the number of uors that a stroked point is allowed to deviate from the original arc. If the -sa flag is specified without a value for arc_str_tol then the software will calculate a reasonable tolerance for you.

[-sb [bspline_str_tol]]:

The -sb flag signifies to include bspline curves and stroke them using the specified tolerance bspline_str_tol. The tolerance indicates the number of uors that a stroked point is allowed to deviate from the original (slow) curve. If the -sb flag is specified without a value for bspline_str_tol then the software will calculate a reasonable tolerance for you. Bspline curves will not be stroked unless this flag is specified on the command line.

[-sc [curve_str_tol]]:

The -sc flag signifies to include curves and stroke them using the specified tolerance curve_str_tol. The tolerance indicates the number of uors that a stroked point is allowed to deviate from the original (slow) curve. If the -sc flag is

specified without a value for curve_str_tol then the software will calculate a reasonable tolerance for you. Curves will not be stroked unless this flag is specified on the command line.

[-U ulf_filename]:

The -U flag precedes the ulf filename (ulf_filename) that identifies the fragmented linework. 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 ulf_filename' or the '-F dgn_filename' are specified on the command line the process will invoke ulfbuilderF. This option should not be used if data is being read directly from a design file using the -F option.

Output Options:

[-Ax]:

This flag specifies the attribute output options. The default (no specification) is to retain all non-duplicate attribute linkages. CAUTION: This option can stack multiple linkages to relational database tables. Other options (x) are:

S : Strip all attribute linkages from the joined linework.
U : Check the linkages and remove duplicate linkages except for feature linkages in the 'feature' table.

For example: -AU

[-b]:

The -b flag signifies to generate bspline curve (type 28) elements - the default is linestring (type 4) elements.

[-C]:

The -C flag signifies to output complex elements for elements containing more vertices than may be stored in one element. Caution: If attributes are present, they will be associated with the complex element header and subsequently lost if the complex element is dropped.

[-c]:

The -c flag signifies to generate curve (type 11) elements - the default is linestring (type 4) elements. Existing curves will not be restroked if curve output is selected. Lines with two vertices will be converted to curves unless they are zero length lines. In this case a zero length line will be output.

[-fdangle]

The -fdangle indicates to flip an element so that it's lone (dangling) endpoint is the last point of the element and the non-dangling point is the first vertex of the element. This is useful for reordering features such as sewer laterals so they always go in the same direction - from a main towards a parcel.

[-fDbValue LinearTableName,DbValueColumn,Operator,flipValue,resetValue]

This flag signifies to flip a linear element if the value stored in DbValueColumn is either less than, less than or equal to, equal to, greater than, greater than or equal to, or not equal to the value indicated by flipValue. Operator is set as either lt, le, eq, gt, ge, or ne based on the intended relationship of the database value to the flipValue.

[-fFrToID LinearTableName,FromId,ToId,orderFlag,breakColumnValue]

The -fFrToID signifies to reorder the linear elements to correspond with the order of the Id values, as designated by the orderFlag. An 'A' signifies ASCENDING order, while any other value signifies DESCENDING order. The breakColumnValue indicates a break in the Id values to separate a prefix from a numeric portion of the Id value(s). A value of '0' indicates to sort based on the entire value.

[-L outlevel [,llev,lsize]]:

The -L flag precedes the output level or level options that follow in the command string. Outlevel should be a value between 1 and 63 and signifies the output level on which the joined linework will be placed. The default is the original level of the element (see option L below). If linework is output to the original level, the original linework will be deleted (the -s option will be overridden).

Outlevel has several other options which are L, F, and O. These options have the following meaning:

L - JOIN BY LEVEL: indicates to consider linework from multiple levels, retain intersections formed by lines of different levels, and place all joined linework on the respective levels of origin. Lines that could be joined but are from different levels will not be joined. This is the default option.

F - JOIN BY LEVEL AND FLAG MULTI-LEVEL LINES: the same as option L, except that ellipses will be placed where multi-level lines cannot be joined. The level and size of these ellipses is indicated by following the F with a level number (llev) and size (lsize) in master units (comma separated).

For example: -L F,62,100

O - OVERRIDE MULTI-LEVEL LINES: the same as option L, except that multi-level lines will be joined and output to the level of the first line segment of the joined multi-level lines.

[-NO]

This flag signifies to process the elements but to not output linear data. It is useful for generating labels without modifying exiting linework.

[-oups]:

This flag signifies to output two(2) vertice lines as a line string (type 4) rather than as a line (type 3) element.

[-PN nodelevel,tablename,idColName,numLinesColName,XcoordColName,YcoordColName]:

This flag signifies to place nodes at the intersection points, including isolated endpoints. Database records will be loaded with info, if table and column info are provided.

[-R]:

The -R flag signifies to reduce (compress) the design file following the completion of segjoin.

[-S s,w,co,cl [,slev,ssize]]:

The -S flag signifies to output linework using the element symbology characteristics that follow in the command string. s,w,co,cl are integer values that represent the element style, weight, color, and class of the output elements. Data items are order dependent and comma separated. If this option is not selected, the default mode is that elements of different symbology will not be joined and

elements will be output with their original symbology.

Several other options may be specified that work similarly to the join by level options except they are join by symbolic property options. These options are s, f, and o. These options have the following meaning:

s - JOIN BY SYMBOLIC PROPERTY: indicates to consider linework symbology, retain intersections formed by lines of different symbolic properties, and place all joined linework on the respective symbolic property of origin. Lines that could be joined but are from different symbolic properties will not be joined. This option may be specified for any one or more of the four symbolic properties of style, weight, color, and class (This is the default option). This option may be specified for a property by simply placing another comma to designate the end of that property's specification (see example below).

f - JOIN BY PROPERTY AND FLAG MULTI-PROPERTY LINES: the same as option s, except that circles will be placed where multi-property lines cannot be joined. The level and size of these circles is indicated by following the f with a level number (slev) and size (ssize) in master units (comma separated).

For example: -S f,,,62,100

o - OVERRIDE MULTI-PROPERTY LINES: the same as option s, except that multi-property lines will be joined and output with the symbolic property of the first line segment of the joined multi-property lines. The s, f, and o options are specified for specific properties of style, weight, color, and class and may be used in combination with numeric override options for other symbolic properties.

For example: -S l,,f,0,62,100

Two other options are available that specify the mode of operation for all the properties. These are F and O and they have the following meaning:

F - JOIN BY PROPERTY AND FLAG MULTI-PROPERTY LINES: this value sets the diagnostic mode for all four properties. Diagnostic circles (differing by color, weight, style, and size) will be placed where multi-property lines are not joined. The level and largest size of these circles is indicated by following the F with a level number (slev) and size (ssize) in master units (comma separated).

For example: -S F,60,50

O - OVERRIDE MULTI-PROPERTY LINES: this value sets the override mode for all four properties. Multi-property lines will be joined and output with the symbolic property of the first line segment of the joined multi-property lines.

For example: -S O

[-SH centroidType,Level[,font,justification,size]]

This flag signifies to convert shapes (type 6 elements) to linework and centroids. Centroids are specified as points(3), textnodes(7), or text(7) to be placed on Level with characteristics of font,justification,size if the centroid is a textnode or text element. Linkages are transferred to the centroid and symbology matches the shape.

[-s]

This flag signifies to save the original linework. This flag can only be used if the resulting linework is output to another level as signified by the -L flag.

[-V num_verts]:

The -V flag signifies that the number of vertices per element should be limited

to the value specified by num_verts. The defaults are the system values of 101 vertices for a linestring and 97 vertices for a curve.

Diagnostic Options:

`[-D dlev,dsize]:`

The -D flag signifies to identify dangling linework. Dangling linework will be identified by placing circles on the specified level (dlev) with a radius equal to dsize (in terms of master units). Data items are order dependent and comma separated. If ellipses (circles) are contained in the element selection criteria along with the linework, they will be considered as pre-existing flags and an additional flag (circle) will NOT be placed at that point.

`[-Jf jLevel,jFlagSize]`

The -Jf flag signifies to flag locations that are potential 'join nodes' if segjoin is allowed to run in a join mode. This flag is used only in a no-join mode, which is automatically set by the process.

`[-L outlevel [,llev,lsize]]:`

This option has level difference flagging capabilities. For specific details see the description under the Output Options Section.

`[-M mLevel,mFlagSize]:`

The -M flag signifies to identify multiple line nodes. Nodes where at least 3 lines meet will be identified by placing circles on the specified level (mLevel) with a radius equal to mFlagSize (in terms of master units). Data items are order dependent and comma separated.

`[-O[F] dist[,level,size]]`

This flag signifies to remove short segments less than the specified distance dist. If the -OF option is selected short segments less than the distance tolerance will be flagged with circles of size placed on level.

`[-S s,w,co,cl [,slev,ssize]]:`

This option has symbolic property difference flagging capabilities. For specific details see the description under the Output Options Section.

Database Update Options:

`[-BD TableName,<newTableName,newTableNameKeyColumn>,]`

This flag indicates ... no duplicates for TableName entity

`[-ID_L2N NodeTableName,NodeIdColumn,LinearTableName,FromIdColumn,ToIdColumn]`

This flag indicates to load linear From/To Ids from the Linear table to the Node Id column in the Node table.

`[-ID_N2L NodeTableName,NodeIdColumn,LinearTableName,FromIdColumn,ToIdColumn,colFormat]`

This flag indicates to load Node Id column values in the Node table into the From/To Id column in the Linear table. From/To values are associated with the Start/End of the linear elements. colFormat is 'N' if the fields are numeric; any other value indicates a character field.


```
[-XY TableName,slopeColumnName,slopeOpt,lengthColumnName,lengthOpt,FromXCol,FromYCol,
  FromZCol,XYZOpt,ToXCol,ToYCol,ToZCol ]
```

where column values to load are optional and governed by the following options:

```
slopeOpt:      3D option
                1 = slope based on Maximum Z - Minimum Z
                2 = slope based on From Z - To Z
                3 = angle based on Maximum Z - Minimum Z
                4 = angle based on From Z - To Z

lengthOpt:     1 = horizontal distance (x,y)
                2 = slope distance (x,y,z) - 3D option

XYZOpt:        1 = load Minimum/Maximum values
                2 = load From/To values
```

Any options involving the Z coordinates or values will only be operational in 3D files.

Data Manipulation and Cleaning Options:

```
[-B<C> breakClass,lowBreakLimit,highBreakLimit,deltaBreakValue,minimumLength ]
```

Indicates to break lines based on an evaluation of the slope of the linear data. This is a 3d option. This argument may be used on the command line up to a maximum of 10 times (10 break classes), in no particular order. The lowBreakLimit of the first class must be set to 0.0. Limits should not overlap. A segment must be at least the length of minimumLength to be classified in that breakClass. Linear data is classified by computing the 'running' average slope of the line from its beginning to the next vertice that may be added to the element. Changes in slope are evaluated, and if an abrupt change is found, of greater than the value of deltaBreakValue for a distance of longer than minimumLength, a change of class will occur and the linear data will be broken. If the -BC option is selected, the broken linework will be output using the color associated with the breakClass with which the linear element is associated.

```
[-f]
```

The -f flag signifies to flip the order of the vertices in the linear elements. If the angle of the line between the first and last vertices of the linear element is > 135° and < 315° (horizontal = 0°) the order of the vertices will be from upper left to lower right; otherwise, the order will be from lower left to upper right.

```
[-fdangle]
```

The -fdangle indicates to flip an element so that its lone (dangling) endpoint is the last point of the element and the non-dangling point is the first vertex of the element. This is useful for reordering features such as sewer laterals so they always go in the same direction - from a main towards a parcel.

```
[-fDbValue LinearTableName,DbValueColumn,Operator,flipValue,resetValue]
```

This flag signifies to flip a linear element if the value stored in DbValueColumn is either less than, less than or equal to, equal to, greater than, greater than or equal to, or not equal to the value indicated by flipValue. Operator is set as either lt, le, eq, gt, ge, or ne based on the intended relationship of the database value to the flipValue.

```
[-fFrToID LinearTableName,FromId,ToId,orderFlag,breakColumnValue]
```

The -fFrToID signifies to reorder the linear elements to correspond with the order of the Id values, as designated by the orderFlag. An 'A' signifies ASCENDING order, while any other value signifies DESCENDING order. The breakColumnValue indicates a break in the Id values to separate a prefix from a numeric portion of the Id value(s).

[-L outlevel [,llev,lsize]]:

This option includes level manipulation capabilities. For specific details see the description under the Output Options Section.

[-O[F] dist[,level,size]]

This flag signifies to remove short segments less than the specified distance dist. If the -OF option is selected short segments less than the distance tolerance will be flagged with circles of size placed on level.

[-S s,w,co,cl [,slev,ssize]]:

This option includes symbolic property manipulation capabilities. For specific details see the description under the Output Options Section.

[-W d_tol,l_tol,a_tol]:

The -W flag signifies to weed the coordinate strings using the specified tolerances. The tolerances are order dependent and expressed in master units (distance and length) and degrees (angle). d_tol is the distance tolerance defined as the distance that a weeded coordinate may be from the resulting weeded line. l_tol is the length tolerance defined as the maximum length between weeded coordinates. a_tol is the angle tolerance defined as the maximum angle that the weeded line deviates from the original line. No coordinates will be weeded if they exceed any one of the specified tolerances. If a value is left out it will be set to zero. Coordinates are saved whose distance, length, or angle values are greater than the specified tolerances (e.g. -W 10,,2.5). This is a 2D option.

To identify zingers enter a minus(-) angle value that will be used to identify a zinger angle tolerance - this tolerance is equal to 180 degrees +/- the angle. Then enter a flagLevel,flagSize for flag placement.

[-WR pixel_size]:

The -WR flag signifies to weed and smooth vectors developed from raster data. Do not use this option on non-raster generated vectors. pixel_size is the pixel size in master units (e.g. 25 for 25 meters) of the raster data from which the vectors were generated. This is a 2D option.

[-WZ]:

The -WZ flag signifies to weed consecutive identical coordinates. This is a 2D option.

[-Z zero_in]:

The -Z flag signifies to identify zero-length lines and handle them according to the value of zero_in. zero_in is the destination level of any zero length lines. Specify a 0 to delete the zero length lines.

Join Options:

[-g]

The -g flag signifies to join linear elements that have different graphic group numbers. The default is to evaluate the graphic group number when joining linework and not join elements of different graphic groups. Graphic group data will be transferred to the joined elements. Linear elements from different graphic groups will have the graphic group number of the first linear component of joined elements.

[-L outlevel [,llev,lsize]]:

This option has level joining capabilities. For specific details see the description under the Output Options Section.

[-N[x] [tablename,column],[NVflagLevel,NVflagSize]]

The -N flag in combination with other options (x) signifies one of several "NO JOIN" options. These options (x) are:

-N

signifies to process all the linework options with the exception of joining the linework. This is a "NO JOIN" flag. It is useful for implementing certain options with joining, such as weeding, removal of zero length lines, and identifying dangling end points without generating "joined" output.

-NA

signifies to join linework with identical attribute linkages (including feature linkages).

-NF

signifies to join linework having identical MGE feature linkages.

-NV [tablename,column],[NVflagLevel ,NvflagSize]

signifies to join linework with identical attribute values as read from tablename,column in the database(e.g. -NV street_info,name to join street of the same name). If NVflagLevel and NVflagSize are set, then segjoin will flag potential join points where the attribute values differ and prevent the linear elements from being joined.

[-S s,w,co,cl [,slev,ssize]]:

This option has symbolic property joining capabilities. For specific details see the description under the Output Options Section.

[-z]

The -z flag signifies to join linear elements that have different zlow/zhigh range values. The default is to consider the zlow/zhigh range values when joining linework and break elements when the zlow/zhigh values change. zlow/zhigh range values will be transferred to the joined elements. If linear elements with different zlow/zhigh values are joined the resulting line will have the zlow/zhigh range values of the first linear component of the new linear element. This flag may be important when working with 2<D design files. This option may also be used to reset zlow/high values to zero; to reset these values place a 0 immediately after the -z (no spaces). For example, -z0 . This is a 2D option.

Label Options:

[-a angleShift]

The -a flag indicates to shift the 'horizon rule' axis by the number of degrees indicated by the value of angleShift(+/-). By default the 'horizon rule' axis

is set at an azimuth of 200 degrees or a bearing of S20E.

```
[-label
tablename,column,level,size[,font,justif,color,weight,angle,minDist,labSpacing,xoffset,yoffset] ]
```

The -label flag signifies to label the lines using database attributes found in a column of tablename. These labels will be placed on level and will have the parameters defined by size, font, justif, color, weight, angle, xoffset, and yoffset. Labels can be placed on only those lines longer than minDist. If labSpacing is specified multiple labels will be placed when the element length is longer than labSpacing.

```
/*
segjoin hints:
*/
```

We recommend that design files be run through the MGE process linecleaner, to remove duplicate linework, before running segjoin. Duplicate linework may result in some unsatisfactory results. 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.

If input is specified using ulf files, the ulf files may be created before running segjoin 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 segjoin 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 segjoin 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 filename. This situation may occur when the process is being run locally, but is accessing files off a server or another node on the network. If segjoin hangs up, the solution is to abort the process, run ulfbuilder... separately, and then run segjoin and specify the -U ulfname option on the command line.

In the event GRS_seg_join 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_seg_join 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 sorted.\$\$, 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 execution of the process. In the event that the process aborts abnormally these work files may need to be removed by the user.

segjoin will process as many lines in the file as there is room to store. This is upwards of 500,000 - 600,000 lines. 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. There is no limit on the number of points that may be in a contiguous string of lines.

```
/*
segjoin examples:
*/
```

segjoin is typically run from the command line. There may be very few arguments to many arguments specified depending on the desired actions of the process. A very

simple application that will join all lines and linestrings (default) within a file (roads.dgn) using elements from all levels (default) is shown below in the first example. The second example uses an element list file for input (lev60.ulf) and directs the process to output all the joined linework as curves on level 40 (-L 40), stroke arcs using the system default tolerance (-sa), and save the original linework (-s). The other examples show how to use segjoin to flag dangling endpoints (-D 63,100), inconsistent level (-L F,62,100) and symbolic property definitions (-S F,60,75), strip attribute linkages (-AS), remove duplicate linkages (-AU), weed vectors generated from raster data (-WR 25), and so forth.

from the UNIX command line:

```
segjoin -F roads.dgn
segjoin -U /usr/mge/%grs/ulf/lev60.ulf -L 40 -s -c -sa
segjoin -U cover.ulf -L F,62,100 -S F,60,75 -D 63,100 -AS
segjoin -F pix_lines.dgn -L 20 -S 0,1,4 -WR 25 -ll 1-4
segjoin -F cover.dgn -ll 1-10 -lw 2 -ls 0-2 -S O -L O -AU -sa 5 -sc 5
segjoin -F covlines.dgn -ll 14-20,30 -L 1 -W 25,,5
segjoin -F covlines.dgn -ll 14-20,30 -L 1 -W 25,,5 -N -R -S 1,1,2
```

for a UNIX shell script application:

```
for i in `cat files.lis`
do
segjoin -F $i -ll 4
done
```

from the WinNT command line:

```
perl -S segjoin.pl -F roads.dgn
perl -S segjoin.pl -U c:\projects\grs\ulf\lev60.ulf -L 40 -s -c -sa
perl -S segjoin.pl -U cover.ulf -L F,62,100 -S F,60,75 -D 63,100 -AS
perl -S segjoin.pl -F pix_lines.dgn -L 20 -S 0,1,4 -WR 25 -ll 1-4
perl -S segjoin.pl -F cover.dgn -ll 1-10 -lw 2 -ls 0-2 -S O -L O -AU -sa 5 -sc 5
perl -S segjoin.pl -F covlines.dgn -ll 14-20,30 -L 1 -W 25,,5
perl -S segjoin.pl -F covlines.dgn -ll 14-20,30 -L 1 -W 25,,5 -N -R -S 1,1,2
```