## Converting Digital Line Graph Data into MGE

Translating DLG data into Intergraph's Modular GIS Environment requires the use of the MGE DLG Translator (MDLG). This is a separate product that is free from Intergraph. It is obtainable from Intergraph's web site. Currently, the product is available at the following address: http://www.intergraph.com/gis/new_free.asp. The product should be downloaded and installed on a machine that currently has Intergraph's MGE already installed.

## How MGE's DLG translator works

The DLG translator product from Intergraph reads DLG-3 Standard or Optional files and translates them into MGE 'features'. The translator reads a user defined parameter file to determine which feature to assign each feature code. In this case, where an MGE project has been created using the SDS/FMS feature names, the parameter file could point to each corresponding SDS/FMS feature. This workflow is acceptable if only an unintelligent graphic representation of the data is to be created. It does not work well for making a smooth translation to the SDS/FMS for the following reasons:

- The translator doesn't handle elements with multiple feature codes in a way that makes a one-to-one correlation possible within the SDS/FMS. For example, if a road feature contains DLG feature codes identifying it as both a **primary highway** and a **bridge**, the MDLG translator will allow the user to assign only ONE SDS/FMS feature code to the resulting graphic. The translator will search the parameter file, find the **first** match (the bridge, for example) and assign the code based on that match. In this case, the highway feature information would be lost.

- The translator doesn't have a facility for handling multiple element types with the same DLG feature code. For example, **tank** features can be represented by points or by areas. This would correspond to the **tank_p** SDS/FMS feature for points or the **tank_c** feature for an area centroid with the surrounding linework becoming the **tank_b** features. The translator doesn't allow the user to make the differentiation between point and area features in the parameter file. This is another reason why the default workflow will not capture all the information.

- The MDLG translator also does not have any facility for updating attributes for features. While certain attribute information (such as highway number, PLSS section number, county fips code, etc.) is stored within the DLG, the parameter file has no facility for allowing the user to assign this information to a particular table and column within the SDS/FMS.

- The MDLG translator will not work with features that span multiple categories. In many instances, SDS/FMS features from several different categories will exist in a single DLG file. For example, the Boundaries DLG may contain data from the **cddod, famgt, bdjur,** and **imrec** categories. The workflow has been modified to allow users to handle this problem.

This workflow will allow the user to translate the data into an appropriate MGE map, then run feature maker based on a query to establish proper feature linkages.

## DLG Import Procedure

This procedure is broken up into sections with an attempt to identify those sections that only occur as setup and those that are used with each translation. Each of these sections will be described in detail afterwards.

Initial Setup

Because a single dlg may contain features that are in several different SDS/FMS 'categories', the DLG translator will have a problem trying to find a 'common category'. To do this, change the feature table being used. In the feature table to be used, change the value of the 'category' field for all features to the same value.

1. Make a copy of the Feature table in the database. Call it feat_dlg.
2. Change the value in the category column of the feat_dlg table to a single value.
3. Add feat_dlg into the mscatalog table
4. Edit the project.tab file so that it points to feat_dlg as the feature table. <project.tab>

Determine which features exist:

5. Use **dlg_anl** to output a report for each dlg to be translated. Optionally, use the **analyze** script to use dlg_anl on a set of files.
6. Parse the report files to get a list of the codes that will be used. Use the DOS 'find' command to extract the information from the report files.
7. Edit the parameter files based on this information. (*.def files).

Translation:

8. Create a seed file.
9. Run the dlg translator on the files. It is best if they are separated by category. The **input** script can be used to translate a group of files.
10. Run the **dlgfmkr** script to feature tag all the necessary features. The dlgfmkr script is a modification of the blkfmkr script delivered with MGE. It reads the parameter file, then runs the MGE command featmkr on all the elements that match a given query.

Graphic Cleanup:

11. Clean the linework. Delete common boundaries, run intersection processor, duplicate line processor, endpoint processor. Make manual edits where necessary.

Graphic Manipulation:

12. For files with areas, create two files: a file with boundaries and centroids, a file with lines, boundaries and points.
13. Complex the area files.
14. Merge the areas back in with the lines and points.
15. Run feature reporter on the file to determine what features are present.
16. <Optional> Update any attribute tables based on existing feature codes using Bulk update.
17. <Optional> Run linkage detacher to delete the dlgin generated feature codes.
18. Use dgnmaker to create the appropriate design files in conformance with the SDS/FMS.
19. Run feature reporter on output files.
20. Run linkage detacher to detach inappropriate linkages from features in final files.
21. Run the chngent.ucm to change the entity number of the files so that they point to the 'feature' table and not 'feat_dlg'.

Undoing setup changes.

22. Edit the project.tab file to point the feature table back to 'feature' instead of 'feat_dlg'.

## Initial Setup

This series of steps is designed to avoid the problem of trying to create features from multiple categories in the same file. MGE has a restriction that multiple features may not exist in the same file. MDLG honors this restriction. In order to avoid the problem, the user will create a 'dummy' feature table which is a copy of the original feature table, only with the category value

changed to a single value. The features will then be linked to this table. After all the processing is finished, the entity number for each element will be changed using a user command so that the features will be linked to the appropriate feature table.

1. Make a copy of the Feature table in the database. Call it **feat_dlg**.

   To do this, use the underlying relational database tools to create a copy of the table. An example using Oracle:

   1. Connect to Oracle using SQL Plus or SQL Worksheet.
   2. Enter the following SQL statement:
   3. Create table feat_dlg as select * from feature;
   4. Execute the statement. A new table will be created that will be an exact copy of the feature table.

   An example using Access:

   1. Open the Access database.
   2. Click on the **Tables** tab.
   3. Click on the **feature** table.
   4. From the Access command menu, select **Edit > Copy**.
   5. From the Access command menu, select **Edit > Paste.**
   6. In the **Paste Table As** box, enter **feat_dlg** as the Table Name: Make sure the button for **Structure and Data** is checked. Select **OK**.

If you are using RIS, use RIS Schema Manager to **include** the **feat_dlg** table. To do this:

1. Open the RIS Schema Manager by selecting **Start > Programs > RIS 05.04** (or current version) **> RIS Schema Manager** from the Windows environment.
2. Use the cursor and click on the name of the appropriate RIS Schema in the **Schema Name** section of the RIS Schema Manager form.
3. Click on the **Data Definition** button on the RIS Schema Manager form. This will open the Data Definition form.
4. Click on the **Include** button on the Data Definition form. This will open the **Include** form.
5. In the **dbms Table Names** section of the Include form, click on the **feat_dlg** tablename.
6. Select **Apply** on the Include form.
7. When the form says 'Table **feat_dlg** included successfully', close all the forms.

2. Change the value in the category column of the feat_dlg table to a single value. This will put all the features into one category. To do this, execute the following SQL command. This can be done using the underlying database (Oracle, Access, etc.), or, if the MGE project is currently using RIS, the RIS Interactive tool can be used.

   ```
   update feat_dlg set category = 200001;
   ```

3. Add the table feat_dlg into the mscatalog table. This can be done by executing SQL commands below. Use the underlying database or the RIS Interactive utility.

   1. Determine an appropriate entitynum for the feat_dlg table. Execute the following command:

      ```
      select * from mscatalog where entitynum < 1000 order by entitynum
      ```

      Use the lowest number that is not currently taken. A number such as **15** will probably be available.

2. Insert a record in the mscatalog table using the following command:

   insert into mscatalog (tablename, entitynum,nextocc) values ('feat_dlg', 15, 1);

4. Edit the project.tab file so that it points to feat_dlg as the feature table. The **project.tab** file is a text file that may or may not exist in your MGE directory structure. If it doesn't, create one using a text editor. Place the file in the main directory of your project, in the level just above the /dgn directory. Add the following line:

   feature:feat_dlg

## Determine which features exist

This step is used to determine which feature codes exist in the DLGs to be translated. By using these commands, the user can pare down the definition files that will be used later in order to save time by not translating features that are not present or are not desired in the final maps. The user can also use these reports to check the quality of the translation to make sure all feature codes are accounted for.

5. Use **dlg_anl** to output a report for each dlg to be translated. Optionally, use the **analyze** script to use dlg_anl on a set of files. Dlg_anl is a command that is included with the MDLG product. To use it:

   1. From MGE, select **Utilities > Command Prompt**. A DOS window will open and the current directory will be set to the home directory of the active project.
   2. From the command prompt, change directories to the location of the DLGs that are to be analyzed.
   3. From the command line type **dlganl –v –F <filename>**. A report of the feature codes that exist in the file will appear on the screen.
   4. To have the analysis output to a report file use **dlganl –v –F <filename> -R <reportfilename>.** The report file will be placed in the **/rpt** directory of the project.
   5. Optionally, the **analyze.cmd** script can be used to analyze a group of files. Analyze.cmd is a perl script that uses the **dir** command to generate a list of files to be analyzed, then runs the dlganl command on them. Open the **analyze.cmd** file with notepad (or other text editor) and edit the line that begins:

      open (DLG, "dir /B *.opt | ")

      Everything between the /B and the | is the list of files that will be processed. If the files end with a .dlg extension, for example, replace *.opt with *.dlg.

      Close the file and type in **analyze** from the command prompt. The report files will be placed in the /rpt directory of the project.

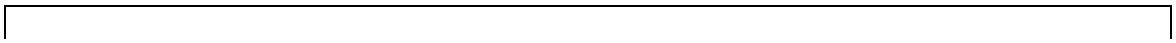
      A sample report file is included below. It is broken into 3 main sections. The first section contains the metadata information about the DLG. This includes the overlay name, scale, and projection information. This information is determines the type of seed file used and the type of parameter file to be used for the translation.

      The second section includes the list of multiple feature code combinations. Typically, this will consist of a code and another code that is often a modifier or attribute. An example of this would be boundary information that contains the State FIPS code.

      The third section gives a list of all the codes that are found within the DLG and their counts. These are broken into Nodes, Areas and Lines. Typically, nodes are translated as point features, areas are translated into centroid features and lines are translated into

line features within MGE. The bounding linework for areas may consist of line feature types AND area boundary feature types. Within the DLG, there are line features that do not have associated feature codes. When these are translated into MGE, they are turned into area boundary feature types. The only way to associate a centroid with its bounding linework is by using MGE's complexer routine, which will be done in a later step.

Header Information:



DLG File Type: Optional

Banner: USGS-NMD  DLG DATA - CHARACTER FORMAT - 09-29-87 VERSION

Overlay name: BOUNDARIES

| Level code: 3 |
| --- |

Scale denominator: 100000

UTM zone code: 15

National Cartographic Unit: NEW ORLEANS, LA
Ground planimetric code: 1 (UTM)
Date of original source and revision: 1983,
Revision date qualifier:
Units of measure code: 2 (meters)
Number of nodes: 12
Number of areas: 5
Number of lines: 15
SW control point (lat/lon): 29.500000 -90.250000
NW control point (lat/lon): 29.750000 -90.250000
NE control point (lat/lon): 29.750000 -90.000000
SE control point (lat/lon): 29.500000 -90.000000

Unique Sets of Multiple Feature Codes:

| Type | Feature Code Combination |
|---|---|
| A | 091-0022 092-0057 |
| | 1 Occurrence(s) |
| A | 091-0022 092-0089 |
| | 1 Occurrence(s) |
| A | 091-0022 092-0051 |
| | 1 Occurrence(s) |
| A | 091-0022 092-0075 |
| | 1 Occurrence(s) |
| L | 090-0201 099-0030 |
| | 1 Occurrence(s) |

Feature Code Counts:

No node feature codes exist in this DLG data file

| Type | Feature Code | Count | Multiple Count | Single Count |
|---|---|---|---|---|
| A | 000-0000 | 1 | 0 | 1 |
| A | 091-0022 | 4 | 4 | 0 |

```
 A        092-0051         1            1               0

 A        092-0057         1            1               0

 A        092-0075         1            1               0

 A        092-0089         1            1               0

Type   Feature Code   Count   Multiple Count  Single Count
----   ------------   -----   --------------  ------------

 L        090-0201         5            1               4

 L        099-0030         2            1               1
```

6. Parse the report files to get a list of the codes that will be used. To get a list of all the feature codes that are used, all of the report files must be opened and read. If there are many files, this may sometimes be laborious. The DOS 'find' command can be used to extract just the lines that contain feature codes. An example would be:

   From the dos prompt:

   find "09" *.rpt

   This will return a list of all the feature codes in all the report files that begin with 09.

7. Edit the parameter files based on this information. There are two sets of parameter files. One is used for the initial translation into MGE. These will not need to be changed. The *.def files are used when taking the translated data and feature tagging it properly. These files will need to be edited in order to save time each time a set of maps will be translated.

   To edit the parameter files:
   1. Select the appropriate parameter file based on the type of DLG. For example, if the DLG category is Hydrography, select the dlghydro.def file to edit.

   2. Open the parameter file using a text editor.

   3. Delete the lines for all codes that are not present in the DLGs. The DLG feature code is the fourth field in each line. Compare it to the parsed reports to determine if that feature is present in the DLG. Note that using the parameter files as-is will not harm the translation, but will result in a lot of error messages because the specified feature code can't be found. The output file will still be created successfully.

      Some features are represented by multiple element types. Because of this, certain feature codes may be present twice or three times. This is to account for features that can be represented as centroids, area boundaries, or lines. Typically these are codes whose minor code is in the 0400 range.

   4. Save the parameter file and close it.

## Translation

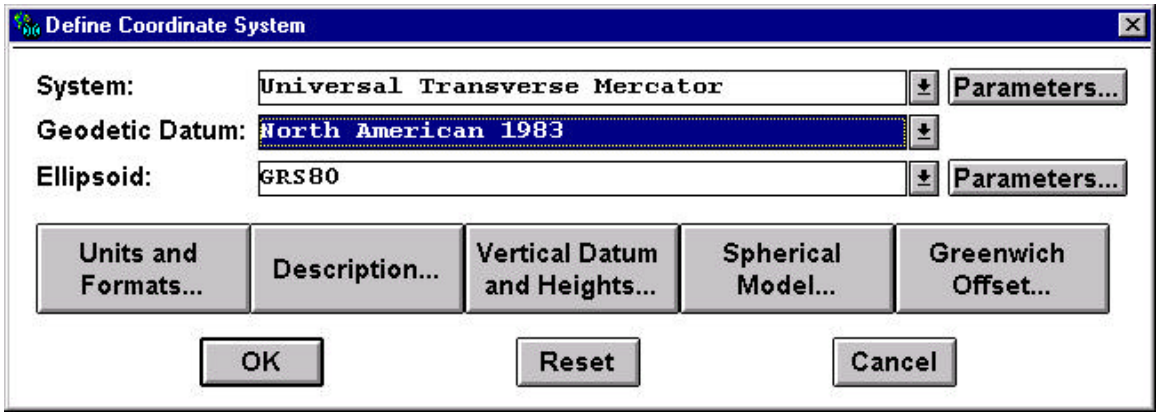
The translation step is when the MDLG translator actually turns the DLG into a map. This is output into a design file with the graphics at their proper geographic location and with default feature tags that are used by the system.

8. Create a seed file. A design file with the proper coordinate system must be used as a seed file. This seed file can be used by ALL dlgs to be translated as long as they all have the same coordinate system and UTM zone, if appropriate. Be aware that typically DLGs of different scales may have different coordinate systems. The **dlgseed.dgn** file has been included as a sample seed file. In most cases it will need to be modified to fit the appropriate zone for translation. To modify this seed file:

    1. Open the dlgseed.dgn file by opening an MGE project and then selecting **Map > Open..** and selecting the dlgseed.dgn file.

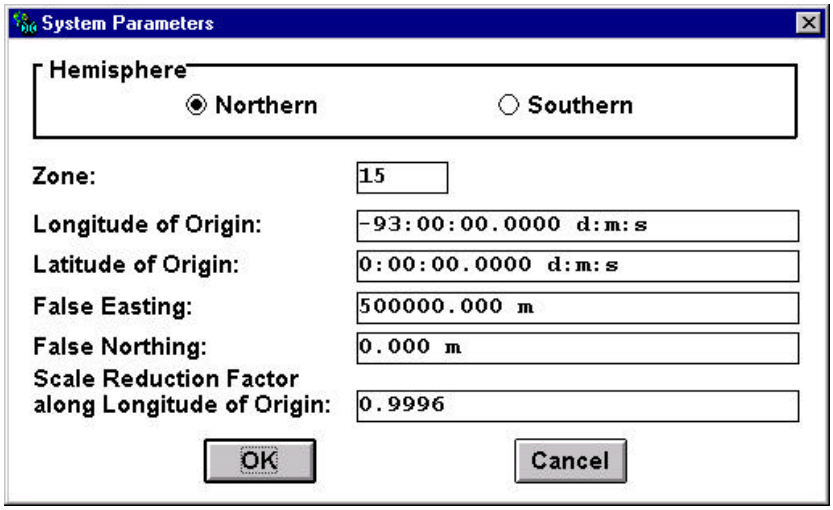
    2. From the MicroStation command menu select **Applications > MGE Coordinate System Operations.**

    3. From the MGE Coordinate Systems Operations menu, select **File > Coordinate System > Primary**. The Define Primary Coordinate System form will appear.




    4. Using the arrow on the right hand side of the System: box, select the appropriate coordinate system.

    5. If the Coordinate System is Universal Transverse Mercator, select the **Parameters…** box to the right hand side of the System: box. The **System Parameters** form will appear.




6. On the System Parameters form, select the appropriate zone.

7. Select **OK** to dismiss the System Parameters box.

8. Select **OK** to dismiss the Define Primary Coordinate System box.

9. Select **File > Exit** to close the seed file.

9. Run the dlg translator on the files. It is best if they are separated by category. The **input** script can be used to translate a group of files. To run the MDLG translator:

   1. Select the appropriate *.gpf file to use as a parameter file. There is one *.gpf file present for each type of DLG. The *.gpf files will force the translator to use default feature names and tables when writing the data to a map and the database. In the next few steps, that data will be used to appropriately tag the features in accordance with the SDS/FMS.

   2. Open an MGE command prompt. Change directories to where the DLG data is stored.

   3. The basic command line for translating one DLG is as follows:

      **dlgin – v –F <filename> -P <parameter file (*.gpf) > -D <output file name> -S <seedfile>**

      For example, if the DLG filename was no2.bdry.opt the command might look like this:

      dlgin – v -F no2.bdry.opt –P bound.gpf –D boundary.dgn –S dlgseed.dgn

      If a series of files from the same category are to be translated into one map, subsequent executions of the command will differ only in that they will not include the seed file parameter. For example:

      dlgin –v –F no3.bdry.opt –P bound.gpf –D boundary.dgn

      It is recommended that an area that is composed of multiple DLGs of the same category all be appended to the same map. This is not required, but it can make graphic cleanup easier.

4. The perl script **input.cmd** has been written to automate translation of a large number of files. Before it can be run, it will need to be modified for the appropriate file names. Open the script with a text editor such as Notepad. Edit the line:

   open (DLG, "dir /B *.opt | ")

   Change the 'dir' statement to more accurately reflect the list of files that are to be translated. This is the same 'dir' statement used in DOS. Change the *.opt portion of the statement so that the appropriate files will be listed. If all files are to be used simply change the statement to:

   dir /B *.* |

   Also, change the name of the parameter file on both lines that begin with **dlgin**. This will be the appropriate .gpf file for the specific category.

   dlgin -v -F $File -P **rail.gpf** -D $Outputfile

   Lastly, change the value of the seed file if it is named other than **dlgseed.dgn**.

   Save and close the script file.

5. To execute the command, copy it to the directory where the DLG files are stored and from the DOS command line type in:

   **input**

10. Run the **dlgfmkr** script to feature tag all the necessary features. The dlgfmkr script is a modification of the blkfmkr script delivered with MGE. It reads the parameter file, then runs the MGE command featmkr on all the elements that match a given query.

    A dlgfmkr parameter file is a text file with a series of lines that has 4 fields. These fields are used to build a temporary list file that is passed to MGE's feature maker process. The fields are separated by colons (:). The fields are, in order from left to right:

    - dlg feature table name: this is the name of the attribute table the MDLG translator attaches to the element that will contain ALL of that elements feature codes.
    - Feature code: this is the name of the default feature code given to elements by the MDLG translator. By default these elements are given names that are a combination of the DLG category (hydro, bdry, etc.) and the MGE feature type (point, centroid, line or boundary).
    - Value in the dall_fcodes field: the **dall_fcodes** field is the column in the dlg feature table name where the translator stores **all** of the feature codes associated with a given element. By searching this field, the script can extract elements that may be given two or more DLG feature codes and attach the appropriate SDS/FMS codes.
    - SDS/FMS feature name: this is the name of the feature within SDS/FMS that the element meeting the preceding criteria will be given.

    The dlgfmkr processes each map file by reading each line within the parameter file, attempting to build a list of elements from the map meeting the criteria, then applying the appropriate SDS/FMS feature name. When each line has been processed, a new map will be output that has the same name as the input file but with a **.fea** extension.

    To run the dlgfmkr script, copy the files dlgfmkr.pl and dlgfmkr.bat into the **dgn** directory of the project. At least one *.def parameter file should be present. Open an MGE command

prompt window and change directories into the **dgn** directory. From the command prompt, type:

dlgfmkr <*.def parameter file> <design file name>

For example:

dlgfmkr dlgroad.def no2road.dgn

In this example, the file no2road.fea will be output.

## Graphic Cleanup

11. Clean the linework. Delete common boundaries, run intersection processor, duplicate line processor, and endpoint processor. Make manual edits where necessary.

    Cleaning the linework is a job that will vary from project to project. The following steps are included as guides to which procedures are commonly necessary.

    **Delete common boundaries**

    If more than one DLG has been appended to a single design file, the boundaries between the two DLGs need to be deleted. This will allow the Endpoint processor procedure to find and fix many dangling endpoints. When these are fixed, elements that are broken at the edge of the DLG can be used to create complex areas without introducing artificial errors such as the DLG border.

    To delete the common boundaries:

    1. Open the design file in MGE.

    2. Place a fence around the common border, as closely as possible.

    3. Set the fence mode to **Clip**.

    4. Delete the fence contents.

    5. Repeat the procedure for any other seams between DLGs.

    **Run Intersection Processor**

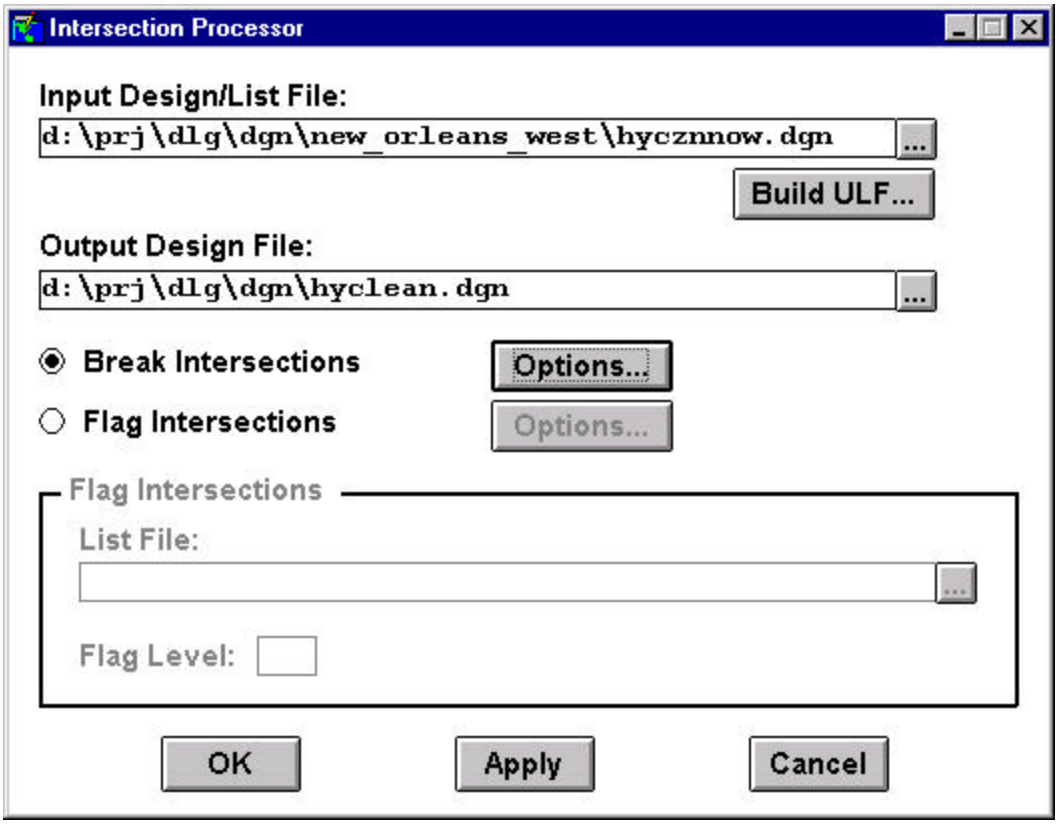
    Intersection Processor will search the map to discover any intersections that are not broken. Having intersections that do not break properly will cause a problem later with the complexing process. In addition, it is simply good practice to ensure that lines break at intersections for topological reasons.

    To run Intersection Processor:

    1. From the MGE form select **Map > Open** and open a map.

    2. From the MicroStation command menu, select **Applications > MGE Base Mapper.** The menu will change to the MGE Base Mapper menu.

    3. From the MGE Base Mapper menu, select **Tools > Linework Processing… > Intersection Processor.** The Intersection Processor form will appear.

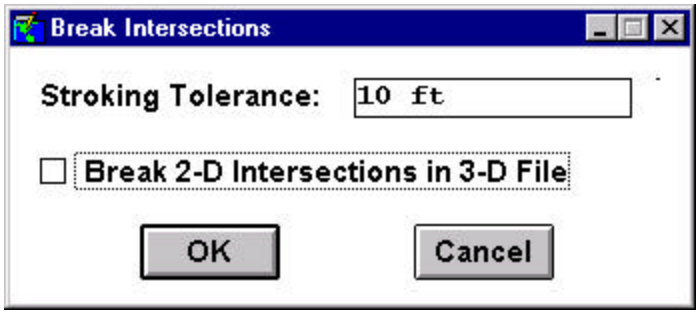



4. On the Intersection Processor form, input the name of the design file to be processed.

5. In the Output Design field, input the name of a file to be output.

6. Click the **Break Intersections** option.

7. Click the **Options…** button besides the Break Intersections option. The Break Intersections form appears.




8. In the Break Intersections box, input a number that represents a stroking tolerance. A smaller number is better.

9. Select **OK** to dismiss the Break Intersections form.

10. Select **OK,** then **Execute and Wait** and **OK** to begin the process. The process will begin and output a new file. Check the new file to make sure lines are broken at their point of intersection.

**Duplicate Line Processor**

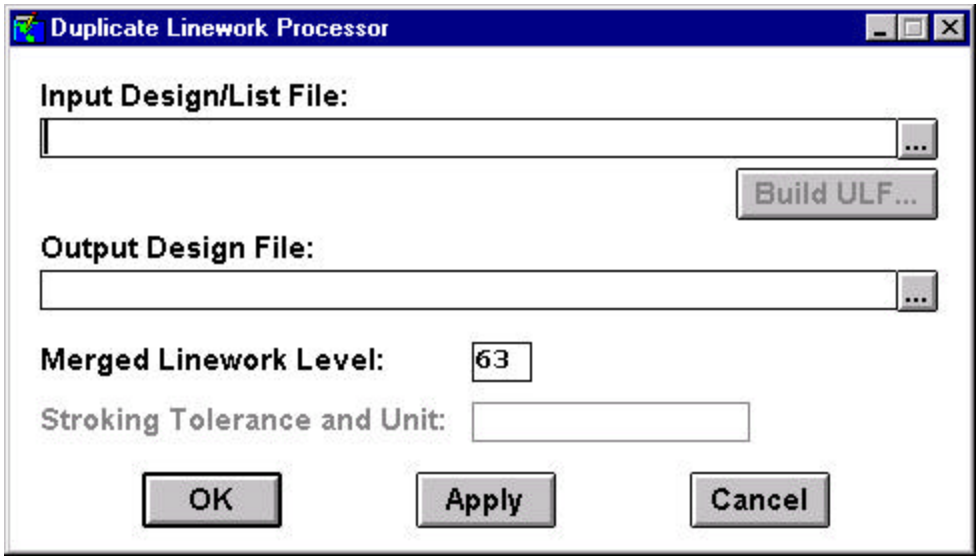
Duplicate lines appear in DLGs and must be deleted. Duplicate lines can cause problems with building complex shapes.

To run Duplicate Line Processor:

1. Open a map in MGE.

2. From the MicroStation command menu select **Applications > MGE Base Mapper**. The command menu changes to the MGE Base Mapper menu.

3. From the MGE Base Mapper menu, select **Tools > Linework Processing … > Duplicate Line Processor**. The Duplicate Line Processor form appears.




4. Input the name of the current design file. Make sure that it is the file output from any previous line cleaning steps.

5. Input the name of an output design file.

6. Select **OK, Execute and Wait** and **OK** to execute the process. All duplicate lines will be merged together on level 63. This doesn't matter because later processes will be selecting features based on their database attributes, not their graphic symbology.

**Endpoint Processor**

The purpose of running endpoint processor is to try to resolve all the free endpoints that exist in the file. Many free endpoints are acceptable. These are mainly lines that simply end at a given point, such as the origin of a stream. Other free endpoints are not acceptable. Mainly

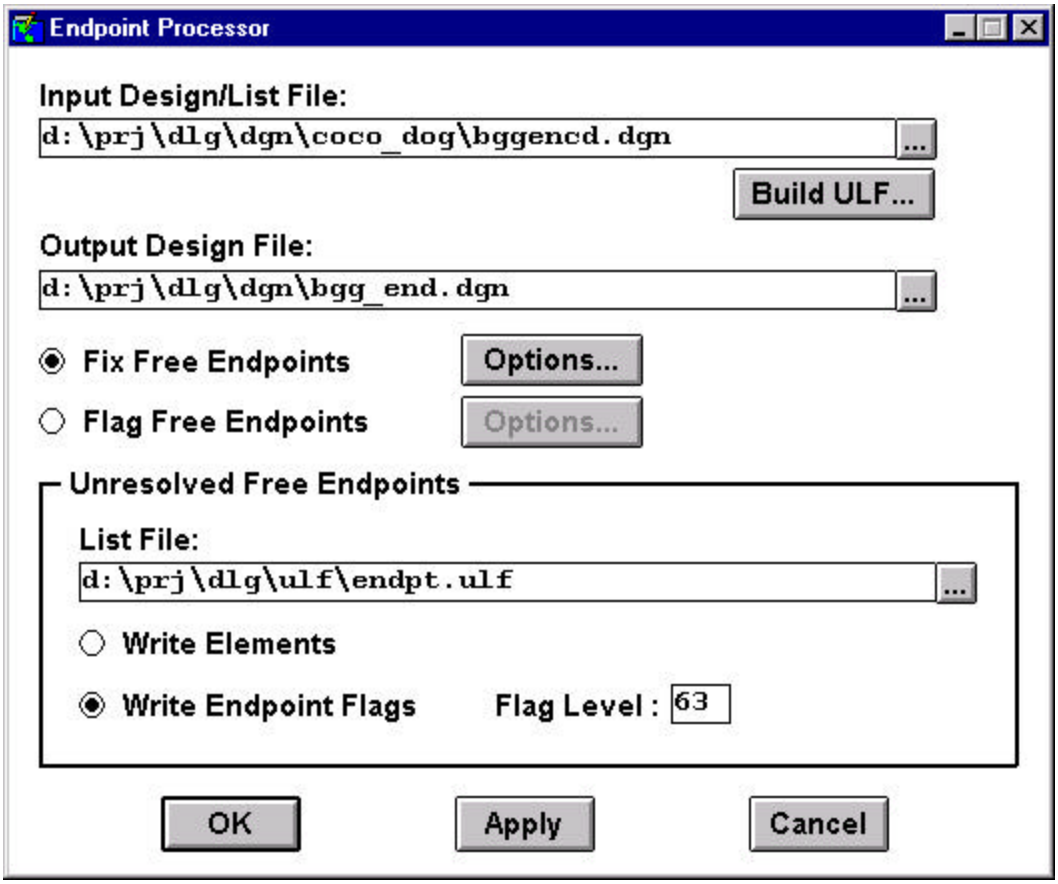
this occurs along the edges of where 2 DLGs are joined. When the boundaries are deleted, many lines remain along the edges to be joined. Endpoint processor joins some of these lines automatically. It flags the remainder for manual cleanup.

To run endpoint processor:

1. Open a map in MGE.

2. From the MicroStation command menu select **Applications > MGE Base Mapper**. The command menu changes to the MGE Base Mapper menu.

3. From the MGE Base Mapper menu, select **Tools > Linework Processing … > Endpoint Processor**. The Endpoint Processor form appears.

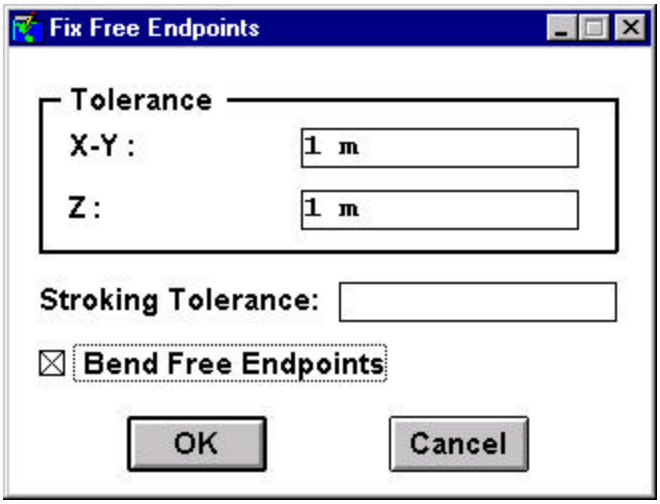



4. Enter the name of the Input file. Make sure it is the same as the output file from the last line processing process.

5. Enter the name of an output file.

6. Click the Fix Free Endpoints option.

7. Click on the **Options…** button beside the Fix Free Endpoints option. The Fix Free Endpoints form appears.




8. Enter **1 m** as the starting X-Y tolerance value. This is the distance of a box that will be drawn to determine if a free endpoint is close enough to another free endpoint.

9. Enter **1 m** as the starting Z tolerance value.

10. Click the Bend Free Endpoints box. This will allow MGE to bend lines where they cross the tolerance boxes so that they can be linked to another free endpoint.

11. Select **OK** to exit the Fix Free Endpoints box.

12. In the Unresolved Free Endpoints section, enter the name of a list file for storing all the endpoint flags for those endpoints that don't get fixed.

13. Select the **Write Endpoint Flags** option. Make the level **63.**

14. Select **OK, Execute and Wait** and **OK**. The process will begin. A new map will be output with corrected linework.

Linework that is not corrected will be flagged. Boxes on level 63 will indicate the endpoints that did not get fixed. Using MicroStation editing tools, go through and link these endpoints to their corresponding match.

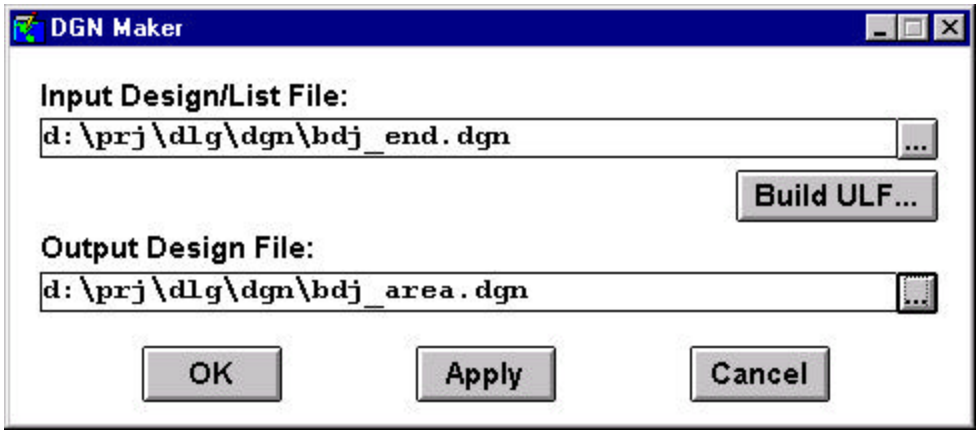
## Graphic Manipulation

12. For files with areas, create two files: a file with boundaries and centroids, a file with lines, boundaries and points. This will allow the complexer process to create the appropriate shapes for area features. To do this, use the **DGN Maker** process within MGE Base Mapper to select the area centroid and boundary features and output them to a new file.

    1. Open a map to be separated.

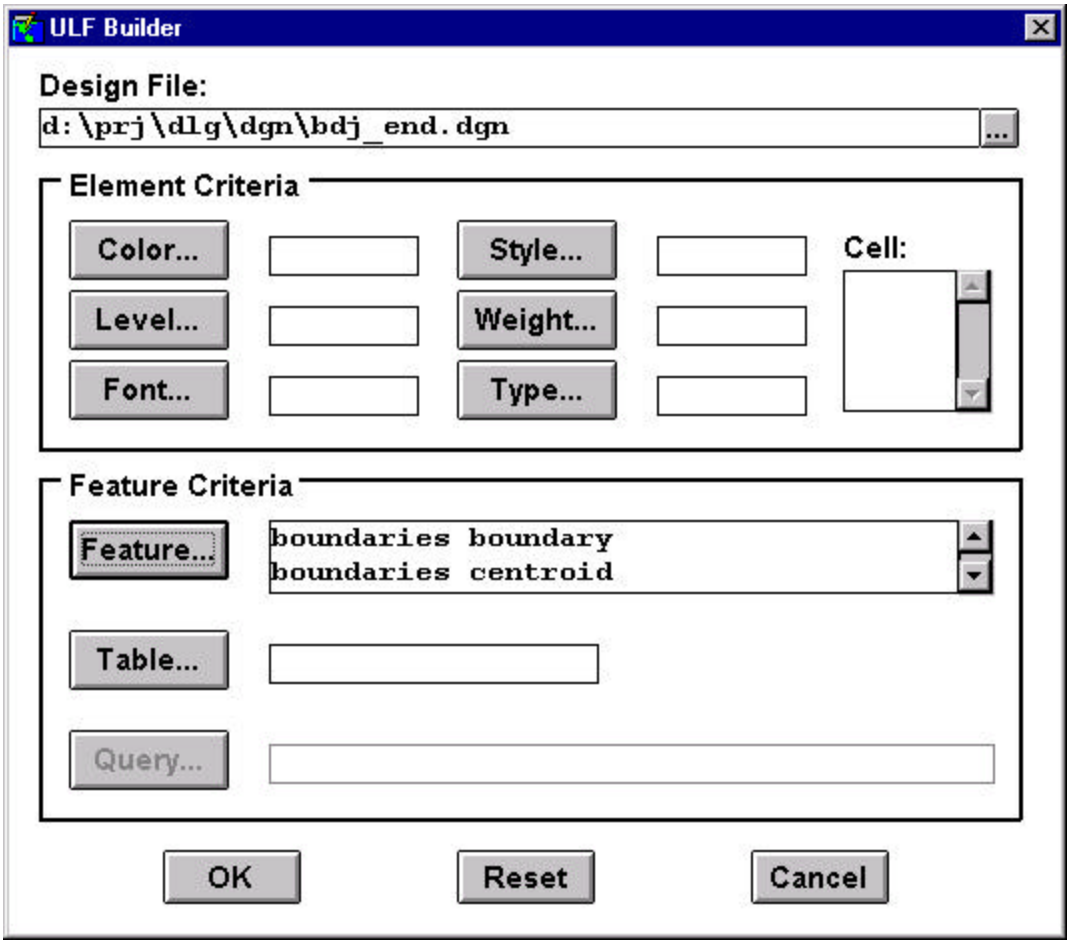
    2. From the MicroStation command menu select **Applications** > **MGE Base Mapper**. The command menu will change.

3. From the MGE Base Mapper command menu select **Tools > Graphics Processing… > DGN Maker…** The DGN Maker form will appear.

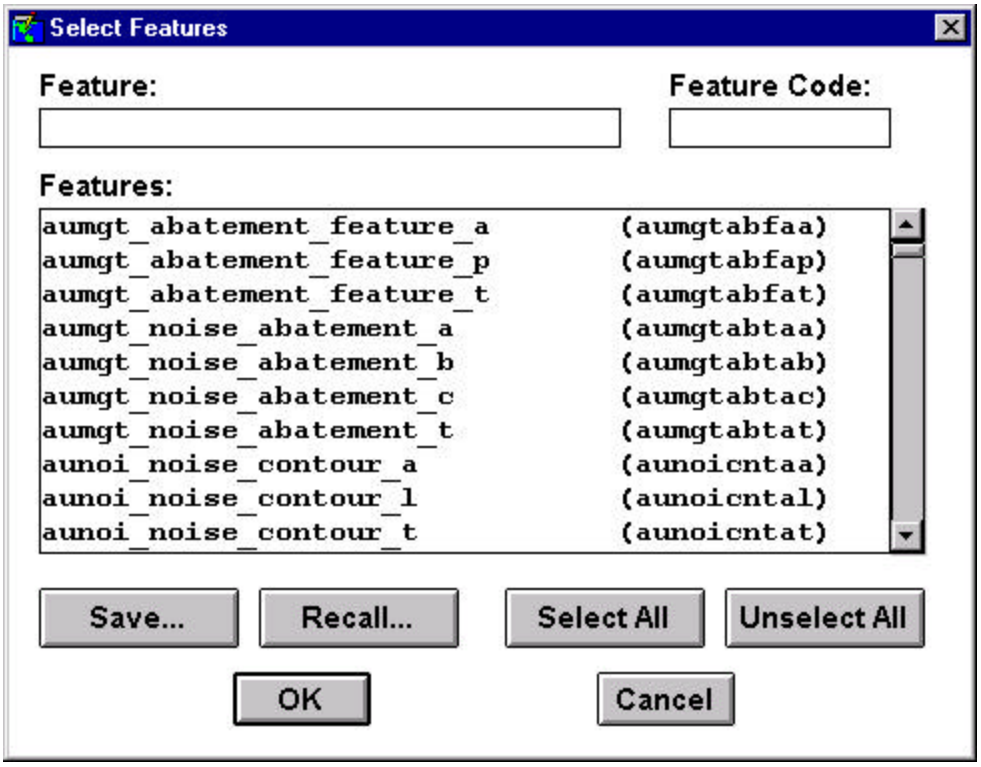



4. Select the most recently edited design file as the Input Design/List File.

5. Click on the **Build ULF…** button. The ULF Builder form appears.



6. On the ULF Builder form, click on the **Feature…** button. The Select Feature form appears.




7. On the Select Feature form, select the appropriate boundary and centroid features. This will change based on the category of dlg you have translated in. For example, for boundary features, the name would be **boundaries boundary** and **boundaries centroid**. This information can be found by using the Feature/Attribute Manager command in MGE Base Mapper and looking at the feature tags for each element.

8. Select the appropriate features and click the **OK** button to dismiss the Select Feature form.

9. Select **OK** to dismiss the **ULF Builder** form.

10. Enter the name of an output design file on the DGN Maker form. Select **OK**. Select the **Execute and Wait** button and select **OK** to begin the process.

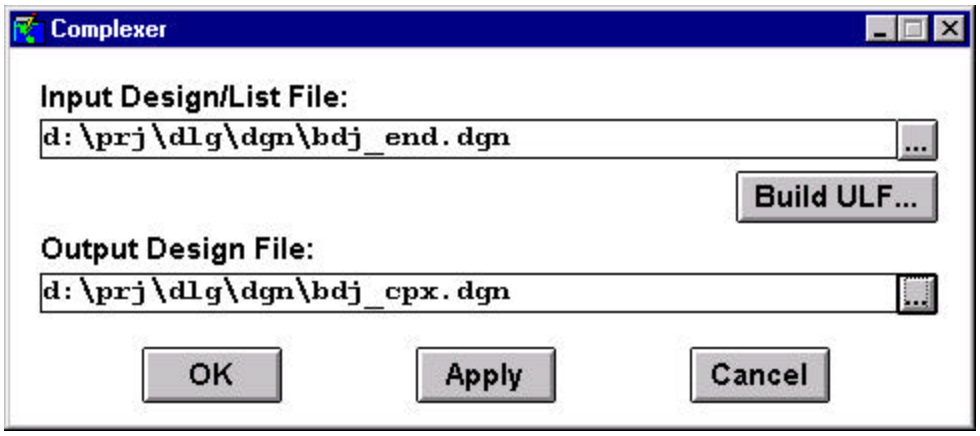
11. When the process is finished, check the new file to make sure there are only centroid and boundary features.

13. Complex the area files. The only way to maintain the geometry of an area defined in a DLG is to complex the areas. To do this, use MGE's Complexer command.

    1. Open the design file created in the previous step.

    2. From the MicroStation command menu, select **Applications > MGE Base Mapper**. The MGE Base Mapper command menu appears.

    3. From the MGE Base Mapper command menu, select **Tools > Graphics Processing… > Complexer …** The Complexer form appears.




4. Enter the name of the file created from the previous step in the Input Design field. This file should contain ONLY the elements that are to be complexed. For this reason, there is no reason to build a list file.

5. Enter an output design file.

6. Select **OK**, Execute and Wait and **OK** to begin the process.

7. When the process ends, check the output file. The centroids should be gone, and replaced with shapes that have links to the centroid table.

14. Merge the areas back in with the lines and points. Merging will put all the elements back into one single file. To merge the files:

    1. Open the original file.

    2. Set the active level to 63. This can be done by keying in: **lv=63** in the MicroStation command menu.

    3. Turn off all the levels in the file. This can be done by keying in: **of=1-63** in the MicroStation command menu and then placing the cursor in the current view and clicking the left mouse button. All the visible graphics should disappear.

    4. Reference the complexed file to the current file. Make sure the complexed shapes are visible.

    5. Place a fence around the complex shapes.

    6. Set the fence mode to **Overlap**.

    7. On the MicroStation keyin box, key in **fence copy**. You will be prompted to enter an origin.

    8. Place the cursor somewhere in the view and press the left mouse button. You will then be prompted to enter a destination point.

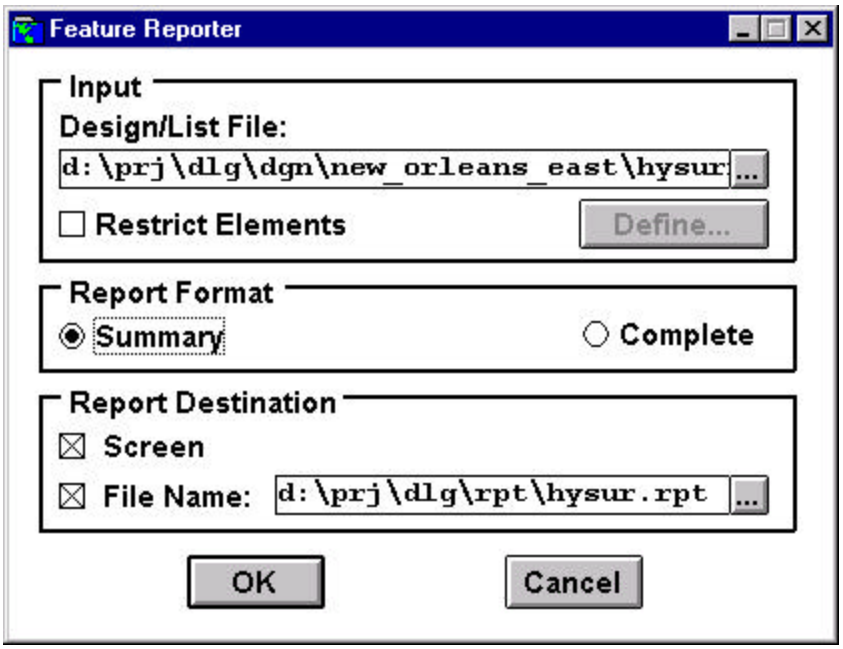
    9. On the MicroStation keyin box, key in **dx=0,0**. This will copy the elements from the reference file into the active file without shifting them in either the x or y direction.

    10. Detach the reference file.

11. Turn on all levels. This can be done by keying in: **on=1-63** and placing a data point in the current view. All graphic elements should appear. If the shapes are missing, go back and review the previous steps.

15. Run feature reporter on the file to determine what features are present. This procedure will help to determine which features currently exist in the design file. By running this process, no features will be overlooked. This will give the user an idea of how many SDS/FMS maps will be created. The first 5 letters of a SDS/FMS feature name identify its category. Within MGE and SDS/FMS, each category must be given its own map. By listing all the features within a file, a list of maps to be created can be generated. To run feature reporter:

    1. From the MGE form, select **Tools > MGE Base Nucleus > Feature Reporter.** The Feature Reporter form appears.




    2. Enter the name of the design file to be analyzed.

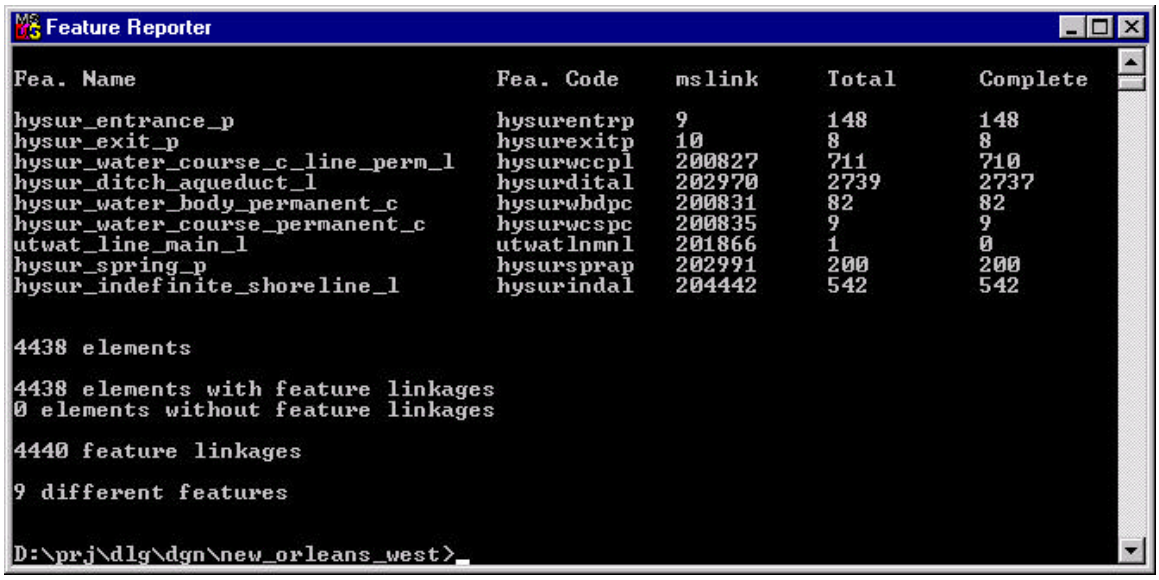
    3. Click the **Summary** button.

    4. The report may either be output to the screen, or to a report file, or both. For most applications, reporting to the screen will be sufficient.

    5. Select **OK** to begin the process.

    6. The process list will appear and state that the process has executed successfully.

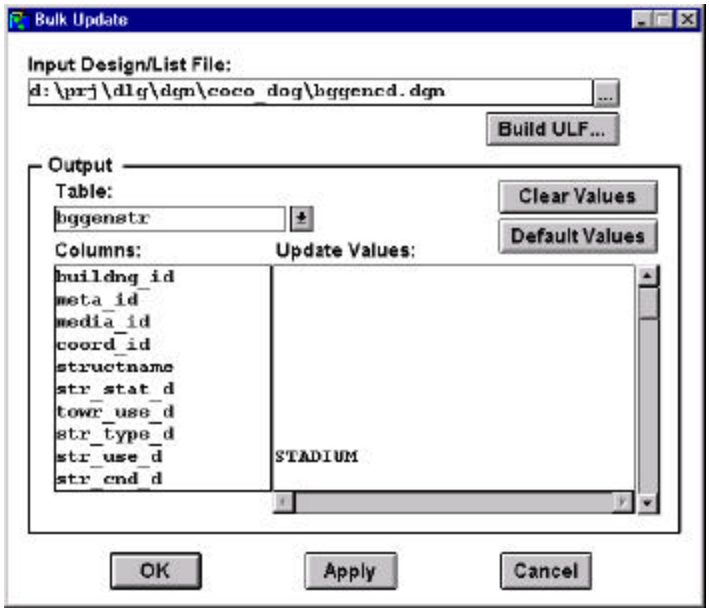
    7. If the report destination is to the screen, a DOS window will appear with information similar to the following:




8. Using the report, decide which SDS/FMS map files will need to be created. It can also be determined if there are features without linkages.

16. <Optional> Update any attribute tables based on existing feature codes using Bulk update. DLG attributes are not easily mapped to SDS/FMS. In some cases, what is considered an attribute in DLG is a feature in the SDS/FMS and vice versa. In order to perform this process, the Bulk Update command must be running, using a list file to identify the features to be updated. An example of how to do this is below:

    1. From the MGE form, select **Tools > MGE Base Mapper > Attribute Processing … > Bulk Update**. The Bulk Update form appears.




    2. On the Bulk Update form, select the design file containing the elements to be updated.

    3. Click on the Build ULF button to bring up the **Build ULF** form.

4. On the Build ULF form, enter the name of the Table to be searched. This will be the default DLG table created for this category. It may be dlg_hydro, dlg_mmade, etc.

5. In the **Query** field, enter the SQL query to select the appropriate DLG fcodes from the dall_fcodes table. For example, to update the fields for a building that is a stadium, enter the following query:

   select dlg_mmade.mslink from dlg_mmade were dall_fcodes like '%200-0456%'

   This will select only those graphic elements who have an fcode that indicates that they are stadiums.

6. Select **OK** to dismiss the ULF Builder form.

7. On the Bulk Update form, select the tablename that will be updated. In the example above, the tablename would be **bggenstr**. The columns of that table appear on the left hand side of the box below.

8. In the **Updated Values** portion of the box, input the value to be updated for each field that will be updated. For example, the value **STADIUM** will be put in the str_use_d field.

9. Select the **OK** and Execute the command. The values will be updated in the table.

17. <Optional> Run linkage detacher to delete the dlgin generated feature codes. Once the features have been linked to their appropriate SDS/FMS feature linkages and all desired attributes have been updated, it is not necessary to maintain the original codes from DLG. These may be deleted using MGE's Linkage Detacher command. Note that this is not a necessary step, but rather a cosmetic one. The linkages can remain without harming further analysis. To detach these linkages:

    1. Using MGE, open a map.

    2. Using the Feature/Attribute Manager command within MGE Base Mapper, identify the feature names and attribute tables to be deleted from the file. Typically these will be the default names assigned to a particular DLG category. For example, for the roads category the following feature names would exist:

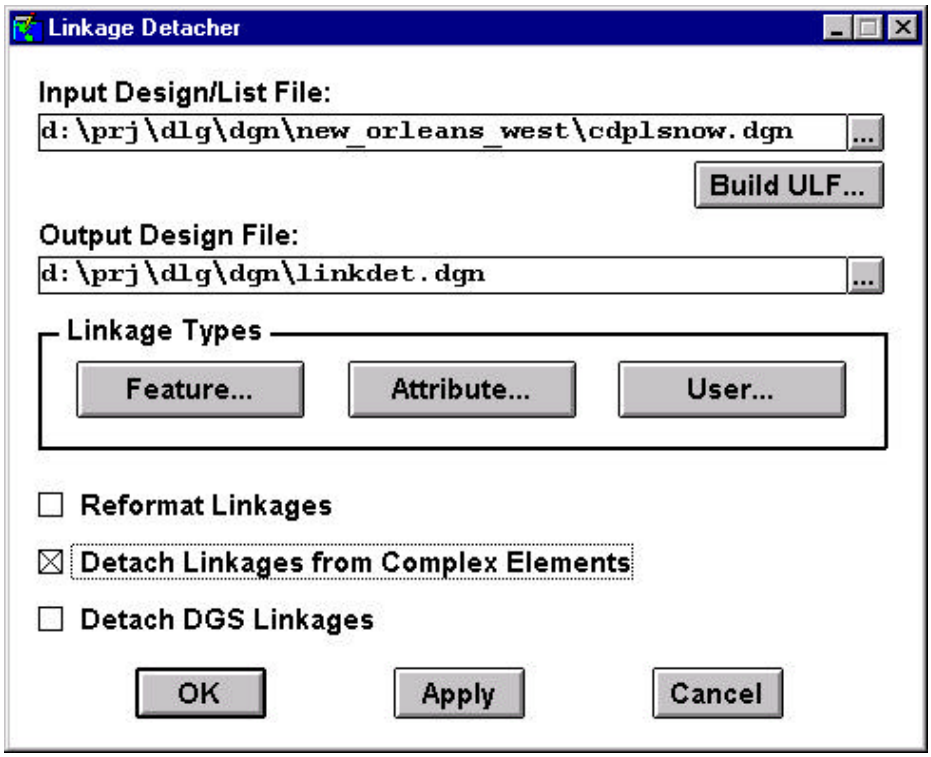
       - road boundary
       - road centroid
       - road line
       - road point

       And the following DLG attribute table would exist:

       - dlg_road

    3. From the MicroStation command menu, select **Applications > MGE Base Mapper**. The command menu will change to the MGE Base Mapper menu.

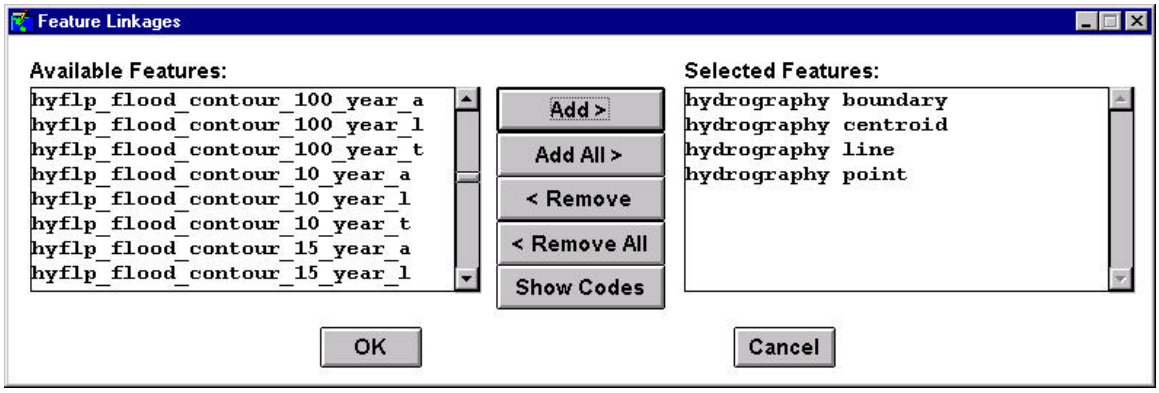
    4. From the MGE Base Mapper menu, select **Tools > Graphics Processing … > Linkage Detacher.** The Linkage Detacher form will appear.




5. Enter the name of the design file to be processed.

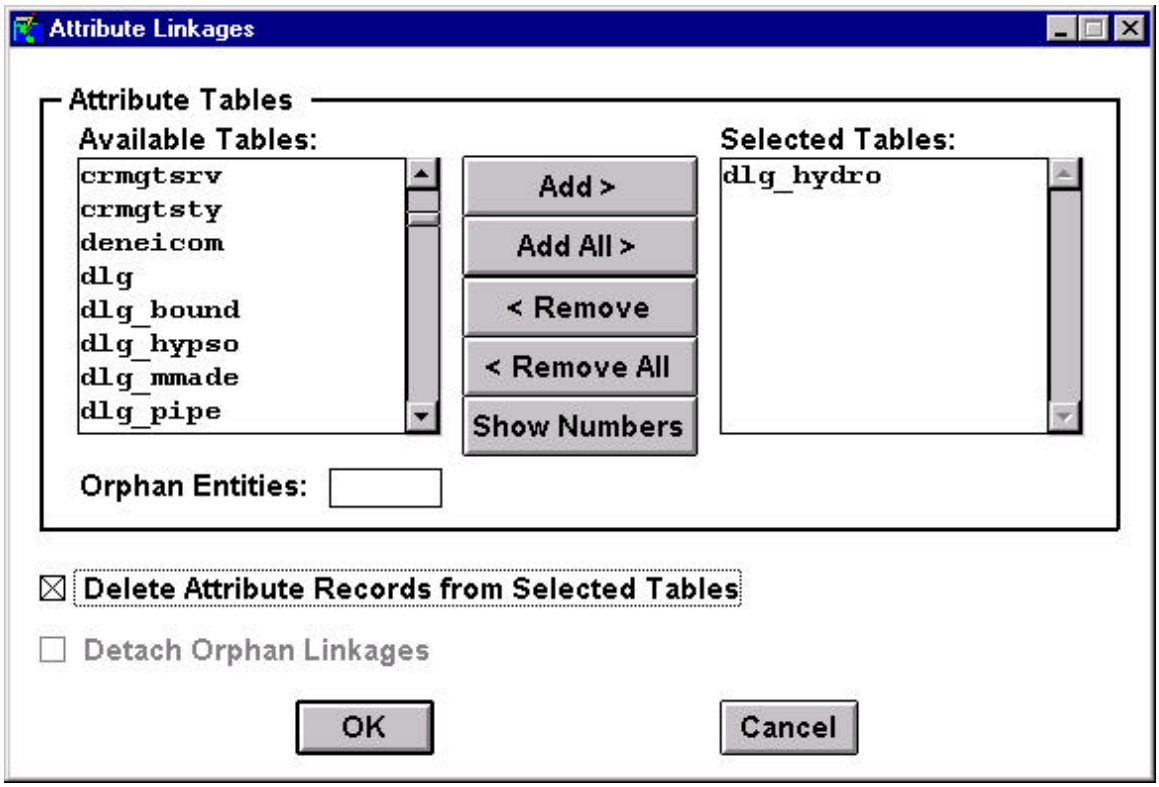
6. Enter the name of the output file to be generated.

7. Click on the **Feature..** button. The Feature Linkages form appears.




8. Select the features to be detached and click on the Add > button to move them to the Selected Features section of the form. Select **OK** to dismiss the form.

9. From the Linkage Detacher form, select the **Attribute…** button. The Attribute Linkages form will appear.




10. Select the attribute table to delete. This table will begin with 'dlg'. Use the Add > button to add the table to the Selected Tables: section of the form.

11. Click on the box beside **Delete Attribute Records from Selected Tables**. This will delete these records from the database. Do not do this if you have further use for the information in the tables.

12. Select **OK** to dismiss the Attribute Linkages form.

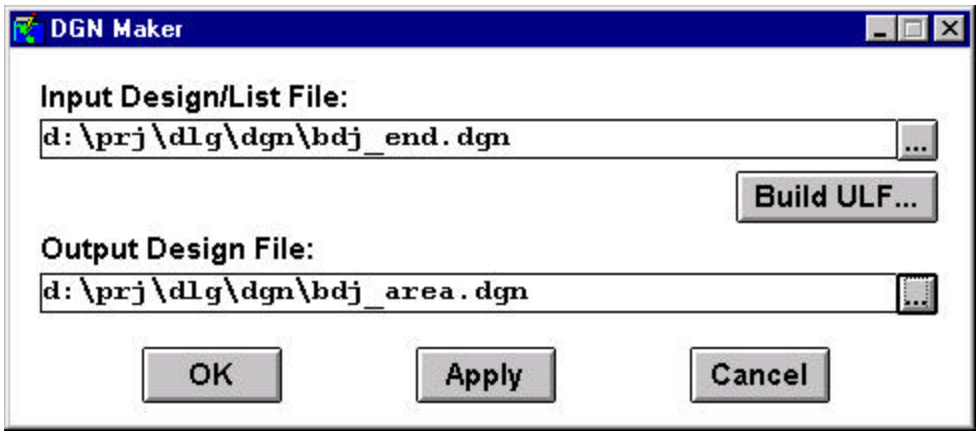
13. On the Linkage Detacher form, select **OK** and execute the command. A new file will be output with linkages stripped.

18. Use dgnmaker to create the appropriate design files in conformance with the SDS/FMS. By using the dgnmaker process, a list file can be built of all the feature codes within a design file that belong to a specific SDS/FMS category. The process will create a design file that ONLY has features that belong to the correct category. This will need to be done once for each category of feature that exists in each design file. Use the output from Feature Reporter from a previous step to determine which maps that will have to be created. To create a file using DGN maker:

    1. Open a map to be separated.

    2. From the MicroStation command menu select **Applications** > **MGE Base Mapper**. The command menu will change.

3. From the MGE Base Mapper command menu select **Tools > Graphics Processing… > DGN Maker…** The DGN Maker form will appear.




4. Input the name of the design file to be segregated.

5. Click on the Build ULF… button to open the Build ULF form.

6. On the Build ULF form, click on the Feature button. Select all the features from the file that are in a common category. This information is present in the report from the Feature Reporter step.

7. Select OK to dismiss the Build ULF form.

8. Enter the output design file name. This should be a SDS/FMS-compliant name. The first 5 characters should be the category abbreviation. The last 3 digits are user defined.

9. Select OK and execute the command. A design file with just the features from a given category will be created.

10. If there are features from more than one category in this original file, run this process again to create maps for all the categories represented.

19. Run feature reporter on output files. This is to check to make sure that the only feature codes that exist in the output files are those that are appropriate for the category. For example, some elements may have been linked, appropriately, to more than one SDS/FMS feature. When the dgn maker process was run, those elements were output to their appropriate files. However, the linkages from the original file remain. These must be deleted.

1. From the MGE form, select **Tools > MGE Base Nucleus > Feature Reporter.** The Feature Reporter form appears.

2. Enter the name of the design file to be analyzed.

3. Click the **Summary** button.

4. The report may either be output to the screen, or to a report file, or both. For most applications, reporting to the screen will be sufficient.

5. Select **OK** to begin the process.

6. The process list will appear and state that the process has executed successfully.

7. Note the features that are present. If there are features that belong in another category, they must have those linkages detached.

20. Run linkage detacher to detach inappropriate linkages from features in final files. This step is used whenever an inappropriate feature linkage is found on an element in a SDS/FMS map. This can occur on elements that have multiple feature linkages that are in differing categories. After running the Feature Reporter step above to discover these problems, use the following steps to detach the inappropriate linkages.

    1. From the MGE form, select **Tools > MGE Base Mapper > Graphics Processing… > Linkage Detacher.** The linkage detacher form appears.

    2. Enter the name of the design file to be processed.

    3. Enter the name of the output file to be generated. Give the output file the same name as the input file. Select **OK** to overwrite the output file.

    4. Click on the **Feature..** button. The Feature Linkages form appears. Add all of the inappropriate feature linkages to the Selected Features area.

    5. Select **OK** to dismiss the Feature Linkages form.

    6. Select **OK** and Execute the command. The output file will be created with the inappropriate linkages deleted.

21. Run the chngent.ucm to change the entity number of the files so that they point to the 'feature' table and not 'feat_dlg'. Up to this point, all the processing has been done using the copy of the feature table created in the setup steps. This was done to avoid the problem of features from different categories being placed in the same file. Once the features have been separated into their appropriate maps, this is no longer necessary. The chngent.ucm is a user command that will change the entity number on each graphic element so that it will point to the feature table and not feat_dlg. To do this:

    1. Using RIS Interactive, or the underlying database, search the mscatalog table to find the entitynumber for the feature table AND feat_dlg. An SQL statement for this would be:

        select tablename, entitynum from mscatalog where tablename like 'feat%';

        This should list both tables. Write down the entitynum value for both.

    2. Using MGE, open a map containing elements that need to be fixed.

    3. From the MicroStation command menu key in:

        uc=d:\prj\dlg\data\chngent.ucm
        (use the appropriate directory path for your system)

    4. The user will be prompted to **Enter the entity number to change**. Enter the entity number for the 'feat_dlg' table.

    5. The user will then be prompted to **Enter new number**. Enter the entity number for the 'feature' table.

6. The user commands will then process all features in the file and change those whose entitynumber meets the criteria.

7. Note that the command will work, but will not provide valid feature linkages until the project.tab file has been changed back to its original state and the project re-initialized.

## Undoing setup changes

22. Edit the project.tab file to point the feature table back to 'feature' instead of 'feat_dlg'. Editing, or deleting this file will force the MGE project to read the default feature table instead of the copy created in early steps. To do this, open the **project.tab** file with a text editor. Change the file back to its original state. The line that begins with feature: should now read:

    feature:feature

    If there was no project.tab file initially, then this file can simply be deleted.