

Geographic Information System (GIS) Database Development, Integration, and Maintenance for the Iowa Department of Transportation Maintenance Division

FINAL REPORT

Sponsored by the Iowa Department of Transportation

December 1998



*Center for Transportation
Research and Education*

IOWA STATE UNIVERSITY



The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the Iowa Department of Transportation.

CTRE's mission is to develop and implement innovative methods, materials, and technologies for improving transportation efficiency, safety, and reliability while improving the learning environment of students, faculty, and staff in transportation-related fields.

**Geographic Information System (GIS) Database Development,
Integration, and Maintenance for the Iowa Department of
Transportation Maintenance Division**

FINAL REPORT

Principal Investigator

Reginald R. Souleyrette

Associate Professor of Civil and Construction Engineering
Associate Director for Transportation Planning and Information Systems
Center for Transportation Research and Education

Research Engineer

Zachary N. Hans

Sponsored by the Iowa Department of Transportation

Center for Transportation Research and Education
Iowa State University
2901 S. Loop Drive, Suite 3100
Ames, Iowa 50011
Telephone: 515-294-5453
Fax: 515-294-0467
<http://www.ctre.iastate.edu>

December 1998

Geographic Information System (GIS) Database Development, Integration, and Maintenance for the Iowa Department of Transportation Maintenance Division

1 Introduction

In 1996 the Maintenance Division participated in several GIS pilot projects as part of the Department's GIS support contract with the Center for Transportation Research and Education (CTRE) at Iowa State University. Based on the success of these pilot projects, the Maintenance Division has committed to utilizing GIS throughout the Division. However, an impediment to integration of GIS within the Division was a lack of familiarity with GIS and its functionality. Potential users within the Maintenance Division require more than a general understanding of GIS as a tool. They must not only feel comfortable with using the software but also understand how they may use GIS as part of their regular functions and activities.

The primary objective of this project was to provide appropriate hands-on GIS training to Maintenance Division staff. The training consisted of three parts: a discussion of general GIS concepts, an introduction to specific software (*Intergraph's GeoMedia*) tools and capabilities, and an overview of management of roadway/roadside feature inventory data within GIS.

Most roadway/roadside features maintained by the Maintenance Division are referenced using mileposts. Mileposts, fractional milepost offsets, and highway names identify where a condition or feature occurs along the roadway. This location information, and associated feature attributes, are maintained in spreadsheet and/or database files. Before this project, no graphic representation of features, and no means by which to create them without manual digitizing, existed. Furthermore, since the location of some features may regularly change, significant effort would be required to maintain manually digitized records. Therefore, dynamic segmentation was identified as a tool that would benefit the Maintenance Division in both their GIS and inventory efforts. Specifically, once a linear referenced control network is established in GIS, features may be dynamically overlaid on the roadway system. Inventory records may be modified in the database and through use of dynamic segmentation, overlaid on the roadway network and displayed in GIS, facilitating better data analysis, presentation, and maintenance.

This document focuses on the procedures necessary to develop and utilize a linearly referenced roadway network to manage roadway/roadside feature inventory data. The procedures outlined in this document apply to MGE Segment Manager 6 and 7, and GeoMedia 2.0. Under these software specifications, all distributed attribute tables, i.e. tables describing features occurring at different locations and extents along the roadway, must exist within a single RIS schema, on a centrally located database. All transportation centers would provide their inventory data to designated individuals at the DOT central office, and the data would then be accessed over the Department's local area network (LAN).

It is important to note that GeoMedia 2.1, scheduled for release in early 1999, includes several enhancements which should better facilitate management of inventory data by the transportation center's themselves. Specifically, GeoMedia 2.1 will allow ODBC connections to distributed attribute tables, multiple distributed attribute tables from multiple databases per connection, and parameter file setup on the GeoMedia seat (allowing the user to connect to any distributed attribute table without running MGE Segment Manager). Therefore, inventory data may be maintained locally in each transportation center (in an ODBC compliant database) and utilized with the appropriate MGE data. Upon evaluation of the software it should become apparent whether the underlying MGE data (control network) must continue to be maintained at the central office and accessed over the LAN, or if a copy of the data may be provided by the central office and stored locally. Non-the-less, the following section outlines the procedures necessary to create a linearly referenced network, given software availability at the time the training sessions were being developed.

2 Database Development

Iowa Department of Transportation (DOT) Maintenance Division inventory data are referenced using a route and milepost offset referencing method. (Data are only collected on the primary highway system.) Data are maintained in attribute tables without a direct relationship to graphic element(s) representing their physical location and extent along a highway. Therefore, a linear referenced highway network is required to allow the overlay and display of these data through use of dynamic segmentation. This section outlines the data and procedures necessary to create the underlying linear referenced GIS database as well as ensure compatibility between the database and Maintenance Division records.

2.1 Create and Populate Distributed Attribute Tables

At the time of the Maintenance staff training sessions, three inventory data sets, collected by the transportation centers, were identified for use within GIS: pavement markings, adopt-a-highway, and access priority ratings. These data sets were provided by several transportation centers and analyzed to determine attributes (and associated data types) located in each data set as well as inconsistencies within and among the data sets. The objective of this process was to establish standard data structures for all data sets, applicable for all transportation centers. This includes specifying attribute fields that must exist within each data set, e.g. the least common denominator among data from all transportation centers, and required attribute fields, and formats, for dynamic segmentation.

Upon examination of the data, several inconsistencies became apparent. The data structures, field names and data types, for the same roadway features, such as pavement markings, were unique to each transportation center. In addition, different formats or software packages were used to maintain the data. (Since the training sessions, the Maintenance Division has standardized on Microsoft products.) Within individual data sets themselves, multiple data types, e.g. numbers and character strings, existed in the same field. In some cases, mileposts were noted as occurring in a county or along a route contrary to the DOT milepost inventory records. A major impediment to dynamic segmentation was the manner in which the locations of features (records) were referenced. Route names were often inconsistent and did not use the same convention as the DOT base record inventory data. Locations were also referenced using fractional milepost values, which can not be used directly for dynamic segmentation, see Appendix A.

Ultimately, data sets representing the same inventory feature type were combined in a single file, in a single format. All data sets were edited to contain the appropriate columns from all transportation centers as well as the required columns for dynamic segmentation and data management. Existing attribute data were also edited to establish consistency, with respect to data type, in each attribute field. These procedures were performed to limit data management and maintenance in the central office. A single database table would exist for each distributed attribute table, containing data from all transportation centers. Since GeoMedia 2.1 should allow for distributed attribute tables to be maintained locally and parameter files to be created locally, this section will not go into detail regarding the attribute fields and data types of each inventory feature. This section will, however, describe how distributed attributed tables are created in Oracle and which attribute fields are required in all attribute tables.

2.1.1 Distributed Attribute Table Minimum Requirements

Inconsistencies in route naming conventions, inaccuracies in milepost referencing and unsupported location referencing limited use of Maintenance Division inventory data directly within GIS. Therefore, several rules were established in an attempt to ensure compatibility between inventory data sets and the underlying GIS database. These rules outline the required columns and formats for all distributed attribute tables using mileposts as a locational reference.

First, the distributed attribute tables must contain a route identifier field. This field is a four-digit number representing an interstate, state (Iowa), or US route. The route identifier must be consistent with the base inventory system, state route field. Routes which are not four-digits in length must possess the appropriate number of leading zeros, e.g. US Highway 20 must be recorded as "0020", Iowa Highway 5 must be recorded as "0005", and US Highway 163 must be recorded as "0163". Because leading zeros may be required, the route identifier field should be a label or character string data type.

Second, the distributed attribute table must contain begin and/or end milepost values. For a linear feature, such as adopt-a-highway locations, both begin and end mileposts are required. For point features, such as signs, only a begin milepost is required. The milepost fields should accommodate a four-digit number, but all four-digits need not be used, i.e. leading zeros are not required. The values contained in the milepost fields should represent the number printed on the face of milepost markers in the field. Because alphanumeric ("A") mileposts may exist, the milepost fields should be a label or character string data type. Fractional milepost values may not be used.

Ideally, the combination of the route identifier and begin/end milepost should identify a single distinct location in the state. However, duplicate mileposts may exist on the same route. Although these occurrences are rare, they result in a non-distinct location identifier. Therefore, an optional, secondary identifier field, associated with the milepost field(s), may also be included in distributed attribute tables. The secondary identifier further refines the location of a given milepost marker. For Maintenance Division applications, it is recommended that the county number be used for the secondary identifier. The field may be either a two-digit numeric or character string field. Leading zeros are not required.

The last required distributed attribute fields are begin and/or end milepost offsets. These fields specify the distance from a feature to a specified milepost marker. The milepost-offset fields should be numeric and support negative values. A positive value will represent an offset towards an increasing milepost marker. A negative value will represent an offset towards a decreasing milepost marker. The same unit of measure is required for offset values in the same distributed attribute table.

2.1.2 Create Distributed Attribute Tables

Because of existing software functionality, all data, including distributed attribute tables, were maintained in an Oracle database. Distributed attribute tables were created using MGE Administrator Feature/Schema Builder. Instructions outlining how to accomplish this follow.

First, new features are created by entering the appropriate information in the *Feature Builder* form. The user should make sure to clear the *Feature Builder* form before creating a new feature. Each feature should represent a different distributed attribute table, e.g. "adopt_a_highway".

Next, the Schema Builder form is cleared and the name of a new attribute table, associated with the newly specified feature, is entered. The *Standard Columns* button is selected to create "mslink" and "mapid" fields. The user then creates the remaining fields using *Column Builder*. Within *Column Builder*, the user enters both the column name and data type. The user may also further refine a column definition by establishing default values, using *Domain Manager*, and allowing/not allowing null values. After entering the desired information, the user should apply each new column. Once all columns have been created, the user must add the table in *Schema Builder*, and add the feature in *Feature Builder*. At this point, a new feature and empty attribute table have been created. In the future, based on GeoMedia 2.1 capabilities, it may be unnecessary to create and maintain all distributed attribute tables in a single location.

2.1.3 Populate Distributed Attribute Tables

Because of existing software functionality, it was necessary to maintain the distributed attribute data in an Oracle database. Since Oracle was not the original software format of these data, distributed attribute data first had to be loaded into the Oracle database. The most efficient means to accomplish this was to convert the distributed attribute data to comma delimited format. Upon creating the comma delimited file, Oracle SqlLoader control files were used to populate the appropriate database tables. A sample control file follows.

```
load data
infile*
into table adopt_a_hwy
fields terminated by','
(st_rte,beg_mpost,beg_offset,beg_co,end_mpost,end_offset,end_co,org,mslink
sequence(max,1))
begindata
0001,2,0.1,89,5,-0.2,89,Boy Scout Troop 89
...
```

After all of distributed attribute tables were populated, the process of creating the control network(s) for dynamic segmentation was initiated. The following procedures are somewhat less dependent on the software versions being utilized and will continue to be applicable until data availability and accuracy warrant change.

2.2 Create Initial Control Network

Intergraph defines the control network as a “set of geographic data that has been structured for segmentation.” Existing primary highway system geographic data (MicroStation format), with associated Base Record Inventory System attributes, provide the framework for creation of the initial control network. This resulting network allows features referenced using the DOT’s linear reference distance point or geographic coordinates (from a GPS receiver, for example) to be overlaid and displayed on the network. Since the inventory data of interest are referenced using milepost location, milepost features (referenced using the linear reference distance point) were placed along this network using dynamic segmentation. The geocoded milepost features were then be used to create a new control network, which in turn was used to overlay inventory data. First, however, the initial control network was established. The following procedures describe how a user would create the initial control network for the first time.

1. Obtain a milepost cross-reference file. This file should contain the four digit route designation (state route number), county number, milepost number (as it appears on the marker in the field), corresponding base record milepoint value (by jurisdiction), and corresponding base record kilometer point value (by jurisdiction). Dave Oesper maintains this file.
2. Create a MGE feature and Oracle table for the milepost data, e.g. “milepost”, including fields for “mslink” and ‘mapid’. Also include fields to note revisions to milepost records, duplicate milepost markers, and milepost marker number. The format of a sample table follows.

Column Name	Null?	Type
MSLINK		NUMBER (10)
MAPID		NUMBER (10)
CO		NUMBER (5)
ST_RTE		VARCHAR2 (4)
MILEPOST		VARCHAR2 (4)
BEG_MIPT		NUMBER (4,2)
BEG_KMPT		NUMBER (6,3)
REV		VARCHAR2 (1)
MILEPOST_NUM		NUMBER (5)
DUP		VARCHAR2 (20)

- Load the milepost data into the newly created table. A sample Oracle SqlLoader control file follows.

```

load data
infile*
append
into table milepost
fields terminated by','
(co,st_rte,milepost,beg_mipt,beg_kmpt, mslink sequence(max,1))
begindata
89,0001,1,1.00,001.609
89,0001,2,2.00,003.219
89,0001,3,3.02,004.860
89,0001,4,4.01,006.453
89,0001,5,5.00,008.047
89,0001,6,6.00,009.656
...

```

- Create a MGE feature and Oracle table for network linear feature control points, e.g. "cp_prim". The format of the table follows.

Column Name	Null?	Type
MSLINK	NOT NULL	NUMBER(10)
MAPID		NUMBER(10)
ST_RTE		VARCHAR2(4)
CO		NUMBER(5)
DIST_3D		NUMBER(6,3)
DIST_2D		NUMBER(6,3)

- Create, or utilize, a MGE project containing the primary highway system, e.g. "primary". A MGE feature representing the primary highway system and Oracle table are required. The primary highway system geographic data (MicroStation format), with associated Base Record Inventory System attributes -- the four digit route designation (state route number), county number, beginning linear reference distance point (in kilometers), lane length (in kilometers), and unique record identifier -- must be associated with each highway feature. A sample table format follows.

Column Name	Null?	Type
MSLINK	NOT NULL	NUMBER(10)
MAPID		NUMBER(10)
CO		NUMBER(5)
JURISD		NUMBER(5)
SYSTEM1		NUMBER(5)
STATUS		NUMBER(5)
RTE_INDICATOR		NUMBER(5)
ST_RTE_PREFIX		VARCHAR2(1)
ST_RTE		VARCHAR2(4)
ST_SEG_SEQ		NUMBER(5)
BEG_LRDPT		NUMBER(10)
BEG_LRDPT_KM		NUMBER(5,3)
LANE LENG		NUMBER(5)
LANE LENG_KM		NUMBER(5,3)
BR_INDEX		VARCHAR2(15)

At this stage in the process, the only graphic data required are features representing the primary highway system. The following steps are based on the assumption that these features exist.

6. Create a parameter file in MGE Segment Manager, e.g. "primary.prm".
 - Define the control point feature table, e.g. "cp_prim".
 - Yes. The distance information restarts...
 - Linear Feature ID: ST_RTE
Secondary ID: CO
Control Point ID: null
3D (true) Distance: DIST_3D
2D (map) Distance: DIST_2D
Distance Unit: km
 - Define the network linear feature table, e.g. "primary".
 - Table Name: primary
Distance Attributes: Distance + Length
Distance Units: km
Linear Feature ID: ST_RTE
Secondary ID: CO
Begin Distance: BEG_LRDPT_KM
Length: LANE LENG_KM
 - Define the distributed attribute linear parameters, e.g. "milepost"
 - Table Name: milepost
 - Reference System: Distance Length (Linear)
 - Linear Feature ID: ST_RTE
Distance: BEG_LRDPT_KM
Secondary ID: CO
Length: LANE LENG_KM
7. Create list (ulf) file containing all "primary" features, e.g. "primary.ulf".
8. Create control network.
 - Linear Element ULF: primary.ulf
Control Point ULF: null
Parameter File: primary.prm
 - Complete Processing
 - Error Report File: primary.rpt
Problem Element ULF: primprob.ulf
 - Linear Element DGN: primary2.dgn
Control Point DGN: cp_prim.dgn
 - (Do not select any other options.)
9. Review primary.rpt and correct any problems in the "primary" graphic or attribute data. Primary.rpt will note which network linear features could not be created, if any, and why they were not created.
(Note: Network topological problems, e.g. overshoots and undershoots, may exist. *Line Cleaner* and/or *MRF Clean* can be used to identify and/or correct such topological problems. If the aforementioned tools are simply used to identify topological problems, manual inspection and modification are required to correct problem areas.)
10. Repeat Steps 7 and 8 until Step 9 reveals no errors.
11. Create list (ulf) file containing all "cp_prim" features, e.g. "cp_prim.ulf".
12. Create coordinate file.
 - Linear Element ULF: primary.ulf
Control Point ULF: cp_prim.ulf
Parameter File: primary.prm

Coordinate File: primary.crd
Report File: primary.rpt

13. Review primary.rpt and correct any problems in the "primary" or "cp_prim" graphic or attribute data. Primary.rpt will note which network linear features could not be created, if any, and why they were not created. Errors may result in creating the coordinate file that were not identified during creation of the control network. However, correcting these problems will entail rebuilding part of the control network.
14. Repeat Steps 7 to 12 until Step 13 reveals no errors.

2.3 Place Mileposts Using Dynamic Segmentation

Milepost marker locations are referenced by longitudinal offsets along the primary highway system (base point offset referencing method). These locations are derived using a combination of video log van mileage and base record mileage. In order to utilize milepost markers as a feature (by which to reference the location of other highway features) graphical representations of the milepost markers must be created in a MGE project. This was accomplished through use of dynamic segmentation.

Upon geocoding the milepost markers as features in the MGE project, they were used as control point features for dynamic segmentation. Because of the nature of milepost markers and control network requirements, virtual mileposts were created to establish a valid control network. Milepost marker locations are referenced relative to the base record inventory system "beg_lrdpt" values, which begin at "0" in each new jurisdiction. Milepost markers are continuous across the state and do not restart at jurisdictional boundaries. However, a valid control network requires routes in each county jurisdiction possess a "0" virtual milepost marker. These "0" virtual milepost markers signify the beginning of a route in the county and occur at milepoint and kilometer point zero (0). A "0" virtual milepost marker may not necessarily be followed by a "1" milepost marker.

It is important to note that the locations of features occurring near the beginning of a route (statewide) are referenced as if a true "0" milepost marker exists. Therefore, virtual mileposts located at the beginning of a route (statewide) must be called "0". Any value may be used for other virtual mileposts, as long as the value has not already been used.

Virtual milepost markers must also be added along portions of routes that contain one or fewer milepost markers, including any "0" virtual milepost markers. Such occurrences are somewhat more difficult to identify without initially placing the milepost markers in the MGE project, defining them as control points in a parameter file, and attempting to create a control network. Creation of these virtual mileposts is discussed in length in the next section.

15. Create virtual "0" mileposts for all distinct routes, by county. This is necessary in establishing the control network required for dynamic segmentation.
 - Identify all distinct state routes, "st_rte", by county, "co" and save the results as an ASCII file. The Oracle sql syntax to select the desired data follows.
 - select distinct "co", "st_rte" from "primary"
 - Using the selection set, create virtual "0" mileposts for each distinct route-county combination. A sample Oracle SqlLoader control file follows. In this example, the milepost field in each record is updated with the value "0". Any value may be used for the milepost as long as it has not already been used, and the record does not represent the beginning of a route (statewide). The beginning of a route (statewide) can be determined using the county sequence, "co_seq", base record attribute.
 - load data
 infile*
 append
 into table milepost

```
fields terminated by','
(co,st_rte,milepost,beg_mipt,beg_kmpt,rev,mmlink (sequence(max,1))
begindata
1,0025,0,0,0,Y
1,0080,0,0,0,Y
1,0092,0,0,0,Y
1,0925,0,0,0,Y
...
```

16. Perform Dynamic Segmentation

- Parameter File: primary.prm
- Coordinate File: primary.crd
- Seed File: seed.dgn
- Output Design File: milepost.dgn
(To Process List and File)
- Report File: primary.rpt
- Features... milepost
- Query... from milepost
- Attributes... Existing Record
- Feature Selection: milepost
- Symbology... No Symbology Defined

17. Review primary.rpt to identify which milepost marker features, if any, were not placed. Determine if the features were not placed because of a problem with the milepost record itself or a problem with the control network.

- If the milepost record is the problem, determine whether it is appropriate to modify the record. If modification of the record does not impact the integrity of the data, make the necessary changes. Otherwise, leave the record unchanged, realizing that these records will not be placed using dynamic segmentation.
- If the control network is the problem, determine where the problem in the network exists, and if it is appropriate to modify the network. If modification of the network does not impact the integrity of the data, make the necessary changes. Otherwise, leave the network unchanged, realizing that some milepost records will not be placed using dynamic segmentation. If changes to the network are made, repeat Steps 7 to 13 as needed.

18. Repeat Step 16, and Step 17, as needed.

As was noted at the beginning of the section, the location of milepost markers longitudinally along the roadway are determined using a combination of videolog van and base record mileage. Geographic coordinates (longitude/latitude) are also available for many, if not all, milepost locations. Given these geographic coordinates, *MGE ASCII Loader* or *Point Placer* may be used to geocode milepost marker locations. These locations may then be used to create the necessary milepost marker control network. However, upon initial comparison of placement of milepost markers with dynamic segmentation and placement by geographic coordinates, many significant discrepancies in marker locations became apparent. Since both of these data sets originated from the same source, one would expect marker locations to nearly coincide. Since they did not, and the discrepancy could not be explained, dynamic segmentation was selected to establish marker locations. In the future, if the accuracy of the geographic coordinates associated with milepost markers can be satisfactorily verified, placement of markers based on coordinate data will likely expedite control network creation.

2.4 Create Control Network Using Milepost Markers

After the milepost markers were geocoded, they were used as control point features. However, additional virtual milepost markers were added along portions of routes that contain one or fewer

milepost markers, including any “0” virtual milepost markers. If virtual markers are not created, the network linear feature can not be created because its direction can not be determined. Specifically, too few milepost features exist along a particular extent of roadway to make its direction (increasing kilometer pointing) apparent. Therefore, virtual milepost(s) must be added to this portion of roadway. The locations where virtual mileposts must be added are not readily apparent. An iterative process of creating the control network and identifying/correcting problem routes (network linear features) that can not be processed is required. Following is a description of this process.

19. Create list (ulf) file containing all “primary” features, e.g. “primary.ulf”.

20. Create list (ulf) file containing all “milepost” features, e.g. “milepost.ulf”.

21. Create a new parameter file in *MGE Segment Manager*, e.g. “primary2.prm”.

- Define the control point feature table, i.e. “milepost”.
 - Yes. The distance information restarts...
 - Linear Feature ID: ST_RTE
Secondary ID: CO
Control Point ID: MILEPOST
3D (true) Distance: DIST_3D
2D (map) Distance:
Distance Units: km
- Define the network linear feature table, e.g. “primary”.
 - Table Name: primary
Distance Attributes: None
Linear Feature ID: ST_RTE
Secondary ID: CO
- Define the distributed attribute linear parameters, e.g. ****
 - Table Name: ****
 - Reference System: Known Marker
What type of distributed attribute data us stored in the database table? Linear
 - Linear Feature ID: ****
Begin Marker ID: ****
Begin Offset: ****
Begin Secondary ID:****
End Marker ID: ****
End Offset: ****
End Secondary ID: ****
 - Distances: ****

AND/OR

- Define the distributed attribute point parameters, e.g. ****
 - Table Name: ****
 - Reference System: Known Marker
What type of distributed attribute data us stored in the database table? Point
 - Linear Feature ID: ****
Marker ID: ****
Offset: ****
Secondary ID: ****
 - Distances: ****

(Note: The distributed attribute tables that the user defines must already exist in the MGE project. See "Create and Populate Distributed Attribute Tables" for details.)

22. Create control network.

- Linear Element ULF: primary.ulf
Control Point ULF: milepost.ulf
Parameter File: primary2.prm
- Complete Processing
- Error Report File: primary2.rpt
Problem Element ULF: primprob2.ulf
- Linear Element DGN: primary3.dgn
Control Point DGN: mpost_cp.dgn
- Allow Movement Of: Linear Elements, Control Points
- Control Point Tolerance Zone Radius: 0.005 mi (or as desired)

23. Review "primary2.rpt" and identify which routes could not be created. Primary2.rpt will note which network linear features could not be created, if any, and why they were not created. Often times, a network linear feature can not be created because its direction can not be determined. Specifically, too few milepost features exist along a particular extent of roadway to make its direction (increasing kilometer pointing) apparent. Therefore, virtual milepost(s) must be added to this portion of roadway.

To determine where along a route a virtual milepost should be added, visual inspection of the network linear features (primary) and control point features (milepost) is required. Many problems in the control network are the result of a change in route designation or a gap in the route. Therefore, a virtual milepost marker record should be inserted at, or near, these locations. Specifically, a new milepost record should be created with a "beg_kmpt" value that occurs before the route designation change or gap. It is usually beneficial to select a kilometer point value that occurs at the beginning or ending of a network linear feature (primary) near these locations. If a milepost feature is missing in the positive direction, it may be appropriate to assign a kilometer point value slightly less than the maximum kilometer point before the route designation change or gap. If a milepost feature is missing in the negative direction, it may be appropriate to assign a kilometer point value slightly greater than the minimum kilometer point after the route designation change or gap.

24. Insert the appropriate records into the milepost feature table. Sample Oracle SqlLoader control file and sql insert script follow. Either method can be used to insert new records into the milepost feature table. Any value may be used for the milepost field as long as it has not already been used. The milepost field, as well as other fields, may initially be loaded with "null" values and later updated.

- load data
infile*
append
into table milepost
fields terminated by','
(mslink,mapid,co,st_rte,milepost,beg_mipt,beg_kmpt,rev,mslink sequence (max,1))
begindata
10291,null,1,'0925',null,null,1.28,'Y'
10292,null,15,'0071',null,null,40.89,'Y'

OR

- insert into milepost (mslink,mapid,co,st_rte,milepost,beg_mipt,beg_kmpt,rev)
values (10291,null,1,'0925',null,null,1.28,'Y');

```
insert into milepost (mslink,mapid,co,st_rte,milepost,beg_mipt,beg_kmpt,rev)
values (10292,null,15,'0071',null,null,40.89,'Y');
```

25. Update mslink values in new records of milepost feature table, ONLY IF “insert” statement was used in previous step. Appropriate Oracle sql scripts follow.
 - select max(mslink) from milepost
 - update milepost set mslink = “results of previous query” + rownum where mslink is null
26. Update beg_mipt, milepost fields for newly created records of milepost feature table. Appropriate Oracle sql scripts follow.
 - update milepost
set beg_mipt=beg_kmpt/1.609347
where rev='Y' and milepost <>'0'
 - update milepost
set milepost=beg_mipt
where rev='Y' and milepost<>'0'
 - update milepost
set milepost=round(beg_mipt,1)
where rev='Y' and beg_mipt>10
27. Repeat Steps 18-20, 22, 24-28 until Step 25 reveals no errors.
28. Create list (ulf) file containing all “milepost” features, e.g. “milepost.ulf”.
29. Create coordinate file.
 - Linear Element ULF: primary.ulf
 - Control Point ULF: milepost.ulf
 - Parameter File: primary2.prm
 - Coordinate File: primary2.crd
 - Report File: primary2.rpt
30. Review primary2.rpt and correct any problems in the “primary” or “milepost” graphic or attribute data. Primary2.rpt will note which network linear features could not be created, if any, and why they were not created. Errors may result in creating the coordinate file that were not identified during creation of the control network. However, correcting these problems will entail rebuilding part of the control network.
31. Repeat Steps 18-20, 22, 24-30 until Steps 25 and 30 reveals no errors.

3 Dynamic Segmentation within GeoMedia

This document has focused on the procedures necessary to successfully create a control network to support dynamic segmentation. However, creating and maintaining the control network is not the responsible of Maintenance Division staff. Maintenance Divisions staff will simply access the appropriate control network

After the control network was successfully established, dynamic segmentation was utilized to overlay and display distributed attributes (inventory data) on the network. This is accomplished by implementing the following procedures.

Within GeoMedia, the user must first open a GeoWorkspace. The user must then connect to an *MGSM (MGE Segment Manager) Warehouse*. This entails specifying the location of the MGSM project (.mge), parameter (.prm), seed (.dgn), and coordinate (.crd) files, and entering the

appropriate RIS schema password. (GeoMedia 2.1 will support ODBC connections in addition to RIS.)

To display and/or query any of the distributed attributes, select New Query from the Tools menu and select the desired feature from the MGSM connection. If the user does not want to display all records of the selected feature, a filter may be defined to refine the query. Spatial overlays may also be performed as well as linear network queries, which are only possible with the distributed attribute tables of the MGSM connection.

The MGSM connection is derived from a MGE project. The user may also want to connect to the underlying MGE project to provide more depth to the distributed attribute displays, specifically the complete primary road and milepost marker coverages. To do so, the user must simply identify the appropriate MGE (.mge) project file and enter a user name, password, and ODBC connect string, and Add Feature Class from the Legend menu.

4 Appendix A: Location Referencing Using Mileposts

1 Creating a Control Network

Most roadway/roadside features maintained by the Maintenance Division are referenced using mileposts. A linear referenced control network must be created to allow the dynamic overlay of these features on the state primary highway system. The linear referenced network will be created using MGE Segment Manager.

A linear referenced control network, representing the primary highway system as of January 1997, was created, and is currently being used, for the Iowa Pavement Management Program (IPMP). The control network was created using base record segmentation, the segment's linear reference distance (kilometer point distance) along a route, and the segment's length. Network control points were automatically placed at the endpoints of each base record segment. Creating a control network in this manner facilitates easier network maintenance, specifically upon changes to base record segmentation and/or attributes, but may result in more computing overhead.

Since the milepost markers do not represent actual milepoint values, the IPMP network, as it exists, can not be used for the Maintenance Division applications. It may be feasible to create a new linear control network using the same underlying network linear features (NLF). However, additional attribute data may be required.

1.1 Milepost-related Issues

- The mileage noted on a milepost marker may not equal the actual mileage (milepoint) along the route.
 - Adjacent milepost spacing may be greater than or less than one mile
 - *Example:* **148 + 0.25** and **149 – 0.75** may not represent the same location.
 - *Note:* The maximum distance between two adjacent mileposts is no more than 1.04 miles. "A" mileposts may exist on a route to ensure that adjacent milepost markers are no more than 1.04 miles apart.
- The difference between the fractional, beginning and ending milepost values associated with a base record inventory segment may not represent the driven distance along the segment.
 - *Note:* Fractional milepost values are calculated using a forward (+) offset from the previous milepost marker. If a previous milepost marker does not exist (which may be the case after the divergence of concurrent routes), a backward (-) offset from the next milepost is used.
- The location of a milepost marker, based on base record mileage, may not be constant over time. Therefore, milepost locations are dynamic with respect to base record mileage.
 - *Note:* The videolog van is used to identify the location of milepost markers along a route. Videolog van distances and base record distances, for the same extent, are compared, and, if not equal, a linear factor is developed and applied to the videolog van mileage to yield the base record mileage. In addition, base record distances that do not fall along the traveled way (usually at interchanges) must also be accounted for in determining the location of mileposts. A milepost equation may be applied.
- Mileposting may be inconsistent along concurrent routes.
 - Mileposting along concurrent routes should be based on the higher priority route (e.g. a US highway has priority over a State highway). If the routes are the same priority, the lower numbered route should take precedence. However, the mileposting along Interstate 35/80, for example, is based on Interstate 80 mileposts (prior to concurrency).
- Milepost markers and route names may change in the field before they are updated in base records.

1.2 Control Network-related Issues

- A control network can not be created using the fractional milepost values. This is because the difference between the fractional, beginning and ending milepost values associated with a base record inventory segment may not represent the driven distance along the segment.
- The IPMP control network was created using base record segmentation. This process requires minimal manual effort and can be easily updated but may increase computing overhead when performing dynamic segmentation.
- The locations of mileposts along IPMP network linear features (NLF) may change annually.
- Control points must exist on a NLF vertex.
 - Mileposts placed using dynamic segmentation will not likely exist at a vertex. NLFs may
 - be automatically split where they intersect control points, but this will result in two graphic elements representing a single base record.
- Milepost locations based on GPS, base record mileage, and other methods may not coincide. In some areas, the difference may be significant.
- Maintenance Offices must reference features using milepost markers and offsets.

1.3 Distributed Attribute-related Issues

- As the physical location of milepost markers change in the field, all distributed attribute records referencing those markers must be updated. If a milepost marker location changes based on its linear or spatial (GPS coordinate) reference, distributed attribute records need not be updated. However, maintenance staff must be aware of the dynamic nature of the milepost marker locations.