SANDIA REPORT
SAND97-1234 • UC-405
Unlimited Release
Printed June 1997

# Software Requirements Specification for the GIS-T/ISTEA Pooled Fund Study Phase C Linear Referencing Engine



Wendy Amai, Juan Espinoza, Jr., David R. Fletcher

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.


Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND 97-1234
Unlimited Release
Printed June 1997

Distribution
Category UC-405

# Software Requirements Specification for the GIS-T/ISTEA Pooled Fund Study Phase C Linear Referencing Engine

Wendy Amai
Advanced Vehicle Development Department

Juan Espinoza Jr.
Decision Support Systems Software Engineering Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1138

David R. Fletcher
Alliance for Transportation Research
University of New Mexico
1001 University Blvd. SE, Suite 103
Albuquerque, NM 87106-4342

## Abstract

This Software Requirements Specification (SRS) describes the features to be provided by the software for the GIS-T/ISTEA Pooled Fund Study Phase C Linear Referencing Engine project. This document conforms to the recommendations of IEEE Standard 830-1984, IEEE Guide to Software Requirements Specification (Institute of Electrical and Electronics Engineers, Inc., 1984). The software specified in this SRS is a proof-of-concept implementation of the Linear Referencing Engine as described in the GIS-T/ISTEA Pooled Fund Study Phase B Summary, specifically Sheet 13 of the Phase B object model. The software allows an operator to convert between two linear referencing methods and a datum network.

# Contents

# Software Requirements Specification for the GIS-T/ISTEA Pooled Fund Study Phase C Linear Referencing Engine

## 1. INTRODUCTION

This Software Requirements Specification (SRS) describes the features to be provided by the software for the GIS-T/ISTEA Pooled Fund Study Phase C Linear Referencing Engine project. This document conforms to the recommendations of IEEE Standard 830-1984, IEEE Guide to Software Requirements Specification (Institute of Electrical and Electronics Engineers, Inc., 1984).

### 1.1 Purpose

This SRS documents the functional requirements for the **Linear Referencing software**. The SRS serves as a statement of understanding between the end users of the software and the software developer of that product. Any desired departures from the specifications described herein should be negotiated by the affected parties.

### 1.2 Scope

Refer to the Pooled Fund Study Phase B Summary binder for a description of the Linear Referencing component.

### 1.3 Definitions, Acronyms, and Abbreviations

Refer to NCHRP 20-27(2) for definitions of transportation components.

All values which are used to describe performance or capacities are underlined.

### 1.4 References

— *IEEE Standard 830-1984, IEEE Guide to Software Requirements Specifications* (Institute of Electrical and Electronics Engineers, Inc., 1984)

— *Geographic Information Systems-Transportation ISTEA Management Systems Server-Net Prototype Pooled Fund Study Phase A Summary*, June 1994.

— *Geographic Information Systems-Transportation ISTEA Management Systems Server-Net Prototype Pooled Fund Study Phase B Summary*, April 1995.

Vanderohe, Alan, et al., "A Report on the Results of a Workshop on a Generic Data Model for Linear Referencing Systems," National Cooperative Highway Research Program 20-27(2), 1994.

## 1.5 Overview

The organization of this SRS conforms to the Prototype SRS Outline given in IEEE 830-1984. The Specific Requirements section follows the Section 3 Prototype Outline 1 given in IEEE 830-1984.

# 2. GENERAL DESCRIPTION

## 2.1 Product Perspective

This product is a proof-of-concept implementation of the Linear Referencing Engine described in the Pooled Fund Study Phase B. The Phase B object model Sheet 13 is the specific component implemented.

## 2.2 Product Functions

This product allows an operator to convert between two linear referencing systems and a datum network.

## 2.3 User Characteristics

Users must be familiar with the basic ideas presented in the Vanderohe NCHRP 20-27(2) paper, specifically regarding the relationship of systems, traversals, reference posts, milepoints, and anchor sections. Users may also be familiar with the Pooled Fund Study Phase B object model, which defines the internal structure of the product.

## 2.4 General Constraints

This product must implement the Location Referencing module as given in the Pooled Fund Study Phase B object model.

This product must run under Microsoft Windows on a PC-compatible hardware platform.

## 2.5 Assumptions and Dependencies

Orientation of links in a traversal is not given. For clarification, see section 3.6.1 Database, regarding table TRAVLINK.DBF.

# 3. SPECIFIC REQUIREMENTS

This section contains the specific software requirements for this product.

## 3.1 Functional Requirements

### 3.1.1 Convert reference post address to datum address

#### *3.1.1.1 Introduction*

A reference post address is a field address consisting of a post number plus an offset from that post. This field address must be converted into a datum address before it can be converted into any other field address. A datum address consists of an anchor section plus an anchor section offset.

#### *3.1.1.2 Inputs*

Reference post address consisting of the following items:

- System ID
- Traversal ID
- Reference post ID
- Reference post offset (this is an arbitrary input from the user)

#### *3.1.1.3 Processing*

This processing is accomplished by TReferenceAddress.ConvertToDatum and DatumSite.Position.

**Procedural description:**

The total link offset is the input reference post offset plus the link offset of the reference post.

Get the ordered list of anchor sections for the link that the reference post is on.

Iteratively subtract off each anchor section length from the total link offset until the total link offset is less than or equal to the current anchor section length. It will be <= the current anchor section length if the reference post address falls on it.

**Pseudo-code:**

Determine which link the reference post is on , and the link offset of the post (TRP-POST.DBF)
Calculate the link offset = input reference post offset + post link offset
Look up list of all anchor sections for this link (LINKSECT.DBF)
For each anchor section for this link (ANCHRSCT.DBF)
    If this is the **first** anchor section on the link
        If this is the **only** anchor section on the link
            If the anchor section goes the same direction as the link
                Total link offset = From node offset + link offset
            Otherwise, the anchor section goes the opposite direction as the link
                Total link offset = From node offset - link offset
            Verify that the total link offset is not greater than the actual link length
            Done.
        Save section information and start processing next anchor section (skip to top of loop)
    If this is the **second** anchor section on the link
        If the anchor section goes the same direction as the link

Total link offset = From node link offset + link offset
Otherwise, the anchor section goes the opposite direction as the link
Total link offset = From node link offset - link offset
If the total link offset is negative, that means the first section was reversed:
Calculate positive total link offset.
If the total link offset is less than or equal to the current anchor section length,
The reference post address has been found.
If the current anchor section goes the same direction as the link
Anchor section offset = total link offset
Otherwise the current anchor section goes the opposite direction as the link
Anchor section offset = current anchor length - total link offset
Done.
Otherwise, the total link offset is greater than the current anchor section length
Subtract off the current anchor section length to prepare for checking next anchor section:
Total link offset = total link offset - current anchor length

#### *3.1.1.4 Outputs*

Datum address consisting of the following items:

- Anchor section ID
- Anchor section offset

### 3.1.2 Convert milepoint address to datum address

#### *3.1.2.1 Introduction*

A milepoint address is a field address consisting of a milepoint distance traveled along a traversal. This field address must be converted into a datum address before it can be converted into any other field address. A datum address consists of an anchor section plus an anchor section offset.

#### *3.1.2.2 Inputs*

A milepoint address consisting of the following items:

- System ID
- Traversal ID

Absolute milepoint along the traversal, currently specified only in miles. A resolution of hundredths of miles is supported.

#### *3.1.2.3 Processing*

This processing is accomplished by TMilepointAddress.GetLink, TMilepointAddress.ConvertToDatum and DatumSite.Position.

**Procedural description:**

Given the traversal, milepoint, and list of links making up this traversal, determine which link the milepoint lands on.

Calculate the link offset of the milepoint.

Get the ordered list of anchor sections for that link.

Iteratively subtract off each anchor section length from the link offset until the link offset is less than or equal to the current anchor section length. It will be <= the current anchor section length if the milepoint address falls on it.

**Pseudo-code:**

Get the ordered list of all links for the input traversal. (TRAVLINK.DBF)

For each link

If the milepoint is greater than the link length,

Subtract the current link length from the milepoint

Following determination of a link and link offset, the pseudo code is identical to that described in the section on converting from a Reference Post address to a Datum address.

#### *3.1.2.4 Outputs*

Datum address consisting of the following items:

- Anchor section ID
- Anchor section offset

### 3.1.3 Convert datum address to reference post address

#### *3.1.3.1 Introduction*

#### *3.1.3.2 Inputs*

Input datum address consisting of the following items:

- System ID
- Traversal ID
- Anchor section ID
- Anchor section offset (this is an arbitrary input from the user)

#### *3.1.3.3 Processing*

Determining a reference post address from a datum address can be described as a two-step process:

1. Given the system ID, traversal ID, anchor section ID, and anchor section offset, determine the corresponding link ID and link offset.
2. Given the link ID and link offset, translate that into a reference post address (reference post ID and reference post offset).

The procedural and pseudo-code descriptions of these steps follows (database tables, .DBF, are listed where used):

1. Determine the link ID and link offset from system ID, traversal ID, and anchor section ID.

**Procedural description:**

Get list of links associated with the input anchor section.
Examine each link(s) with matching system and traversal until link and link offset are found.
Traverse each anchor section associated with this link, looking for the input anchor section ID
Meanwhile, accumulate the link offset until the anchor section ID is found
When the anchor section is found, finish calculating the link offset

**Pseudo-code:**

For each link associated with the input anchor section ID, do the following: (LINKSECT.DBF)
Find a link that has matching system ID and traversal ID. (TLINK.DBF)
Populate the associated Tlink object and its FromNode and ToNode objects (TNODE.DBF)
For each anchor section associated with this link, do the following: (LINKSECT.DBF)
Get the endpoints and length for this anchor section. (ANCHRSCT.DBF)
If this is the **first** anchor section of this link, try to place the datum address on it:
If this is the **only** anchor section of this link
Verify that the anchor section ID matches the input anchor section ID
Verify that the anchor section ID matches the From and To node anchor section ID's
Verify that the input anchor section offset falls between the node anchor offsets
If the From node offset < To node offset
Link offset = input anchor section offset - From node offset
If the To node offset < From node offset
Link offset = To offset - input anchor section offset
Done with determining link ID and link offset
Otherwise this is not the only anchor section of this link:
Save this anchor section's ID, end points, and length.
If this is the **second** anchor section of this link, determine if datum was on first link:
If input section ID matches first anchor section ID, determine first section orientation:
If second section connects to the head of the first section
Verify input section offset falls on first section (Input offset < From node offset)
Link offset = input anchor section offset - From node offset
Done with determining link ID and link offset
If second section connects to the tail of the first section
Verify input section offset falls on the first section (Input offset > From node offset)
Link offset = From node offset - input anchor section offset
Done with determining link ID and link offset
If input section ID does not match first anchor section ID, update accumulated link offset:
If second section connects to the head of the first section
Accumulated link offset = First section length - From node offset
If second section connects to the tail of the first section
Accumulated link offset = From node offset
If this is the **third or greater** anchor section of this link, determine if datum on previous link:
If input section ID matches previous section ID, determine previous section orientation:
If current section connects to the head of the previous section
Link offset = Accumulated link offset + input anchor section offset
Done with determining link ID and link offset

If current section connects to the tail of the previous section

Link offset = Accumulated link offset + (previous section length - input offset)

Done with determining link ID and link offset

If input section ID does not match previous section ID, update total length of link so far:

Accumulated link offset = accumulated link offset + previous anchor length

If the current anchor section ID matches the input anchor section ID, datum is on current section

If head of current section connects to previous section

Verify input anchor section offset <= To node offset

Link offset = Accumulated link offset + input anchor section offset

Done with determining link ID and link offset

If tail of current section connects to previous section

Verify input anchor section offset >= To node offset

Link offset = Accumulated link offset + (current section length - input section offset)

Done with determining link ID and link offset

2. Translate link ID and link offset into reference post address, which consists of a reference post ID and a reference post offset.

**Procedural description:**

Get the ordered list of reference points for this link (list of reference posts for this link is ordered by link offset, from least to greatest).

Determine the closest reference post in the negative direction to the input link offset by examining each post's link offset until a post's link offset is found that exceeds the input link offset. Then the closest post in the negative direction is the immediately previous post.

Determine the reference post offset by subtracting the post's link offset from the input link offset, or if the initial post's link offset already exceeds the input link offset, then the post offset will be negative.

**Pseudo-code:**

For each reference post on this link, do the following: (TRP_POST.DBF)

If this post's link offset is greater than the input link offset, then

If this is the first post on this link

Post ID = Current post ID

Post offset = Link offset - Post link offset (negative quantity)

If this is not the first post on the link

Post ID = Previous post ID

Post offset = Link offset - Previous post link offset

### *3.1.3.4 Outputs*

Reference post address consisting of the following items:

- Reference post ID
- Reference post offset

### 3.1.4 Convert datum address to milepoint address

#### *3.1.4.1 Introduction*

#### *3.1.4.2 Inputs*

Input datum address consisting of the following items:

- System ID
- Traversal ID
- Anchor section ID
- Anchor section offset

#### *3.1.4.3 Processing*

Determining a milepoint address from a datum address can be described as a two-step process:

1. Given the system ID, traversal ID, anchor section ID, and anchor section offset, determine the corresponding link ID and link offset.
2. Given the link ID and link offset, translate that into a milepoint address along the input traversal.

Step 1 is identical to that in the previous section, 3.1.3 Convert datum address to reference post address. Consult that section for details on Step 1.

Step 2 is as follows:

Translate link ID and link offset into milepoint address, the absolute milepoint along the input traversal.

**Procedural description:**

**Pseudo-code:**

#### *3.1.4.4 Outputs*

Milepoint address consisting of an offset which indicates absolute milepoint along the input traversal.

## 3.2 External Interface Requirements

### 3.2.1 User Interface Requirements

The interface will be Microsoft Windows-based, but the linear referencing data will be presented textually rather than graphically.

### 3.2.2 Hardware Interface Requirements

None.

### 3.2.3 Software Interface Requirements

ARCinfo requires milepoint data, but the exact interface is still to be determined.

### 3.2.4 Communication Interface Requirements

None.

## 3.3 Performance Requirements

Completion of all calculations should not take longer than 60 seconds.

## 3.4 Design Constraints

### 3.4.1 Standards Compliance

This product uses the standard terminology given in NCHRP 20-27(2) and the Pooled Fund Study Phase A and Phase B summaries.

### 3.4.2 Hardware Limitations

None.

## 3.5 Attributes

### 3.5.1 Security

N/A

### 3.5.2 Maintainability

N/A

### 3.5.3 Availability

N/A

### 3.5.4 Transferability/Conversion

When moving this product to another platform, the path for the database alias LRMtest must be changed to reflect the actual location of the LRM test databases. This can be done either through the Delphi Database Engine Configuration or through the Delphi Database Desktop.

## 3.6 Miscellaneous Requirements

### 3.6.1 Data Base

The following database tables must exist in order for this product to function. They must use the exact names and structure as given below. All files are in dBASE IV format. All files are unordered (records are in no particular order).

Because of naming constraints, no more than 99 each of anchor points, anchor sections, links, nodes, traversals, reference posts, or milepoints are supported, with names 'X01' through 'X99', where 'X' is one of 'A', 'S', 'L', 'N', 'H', 'R', and 'M' respectively.

Filename: **ANCHORPT.DBF**
Contents: All anchor points.

| Field name | Type | Size | Notes or example |
|---|---|---|---|
| ANCHOR_PT | C | 3 | 'A02' |
| DESCR | C | 65 | |
| LOCATION | C | 65 | GPS location string |

Filename: **ANCHRSCT.DBF**
Contents: All anchor sections.

| Field name | Type | Size | Example or notes |
|---|---|---|---|
| SECTION_ID | C | 3 | 'AS3' |
| FROM_PT | C | 3 | 'A02' |
| TO_PT | C | 3 | 'A03' |
| DISTANCE | N | 6:2 | 0.04 |
| UNITS | C | 10 | 'Miles' |

Filename: **LINKSECT.DBF**
Contents: All links and the one or more associated anchor sections.

| Field name | Type | Size | Example or notes |
|---|---|---|---|
| LINK_ID | C | 3 | 'L07' |
| SECTION_ID | C | 3 | 'AS3' |
| ORDINAL | N | 2:0 | 3 |

Filename: **SYSTEM.DBF**
Contents: All systems.

| Field name | Type | Size | Example or notes |
|---|---|---|---|
| SYSTEM_ID | C | 3 | 'S02' |
| CLASS | C | 10 | 'SR' |
| NAME | C | 65 | 'State roads' |

Filename: **TLINK.DBF**
Contents: All links.

| Field name | Type | Size | Example or notes |
|---|---|---|---|
| LINK_ID | C | 3 | 'L07' |
| FROM_NODE | C | 3 | 'N06' |
| TO_NODE | C | 3 | 'N07' |
| WEIGHT | N | 6:2 | 1.36 |
| UNITS | C | 10 | 'Miles' |
| SYSTEM_ID | C | 3 | 'S02' |
| TRAVERSAL | C | 3 | 'H03' |
| LINK_OPEN | L | | True |

Filename: **TNODE.DBF**
Contents: All nodes.

| Field name | Type | Size | Example or notes |
|---|---|---|---|
| NODE_ID | C | 3 | 'N36' |
| ANCH_SECT | C | 3 | 'AS5' |
| ANCH_OFFS | N | 6:2 | 3.51 |
| DESCR | C | 65 | '?' |
| NODE_OPEN | L | | True |

Filename: **TRAVLINK.DBF**
Contents: All traversals and the one or more associated link.

| Field name | Type | Size | Example or notes |
|---|---|---|---|
| TRAV_ID | C | 3 | 'H45' |
| LINK_ID | C | 3 | 'L22' |
| ORDINAL | N | 2:0 | 8 |

Filename: **TRP-MILE.DBF**
Contents: All milepoints.
Note -- This file is superfluous, as the From Node on each link always milepoint 0.

| Field name | Type | Size | Example or notes |
|---|---|---|---|
| MPT_ID | C | 3 | 'M40' |
| TRAVERSAL | C | 3 | 'H17' |
| LINK_ID | C | 3 | 'L26' |

Filename: **TRP-POST.DBF**
Contents: All reference posts.

| Field name | Type | Size | Example or notes |
|---|---|---|---|
| POST_ID | C | 3 | 'R52' |
| DESCR | C | 65 | 'SR-15 on and off ramps' |
| TRAVERSAL | C | 3 | 'H11' |
| LINK_ID | C | 3 | 'L14' |
| LINK_OFFS | N | 6:2 | 1.45 |

Filename: **TRV-HWY.DBF**
Contents: All highway traversals.

| Field name | Type | Size | Example or notes |
|---|---|---|---|
| TRAV_ID | C | 3 | 'H12' |
| SYSTEM_ID | C | 3 | 'S01' |
| NUMBER | C | 10 | '114' |
| NAME | C | 65 | 'Central Avenue' |
| DIRECTION | C | 1 | 'W' |

### 3.6.2 Operations

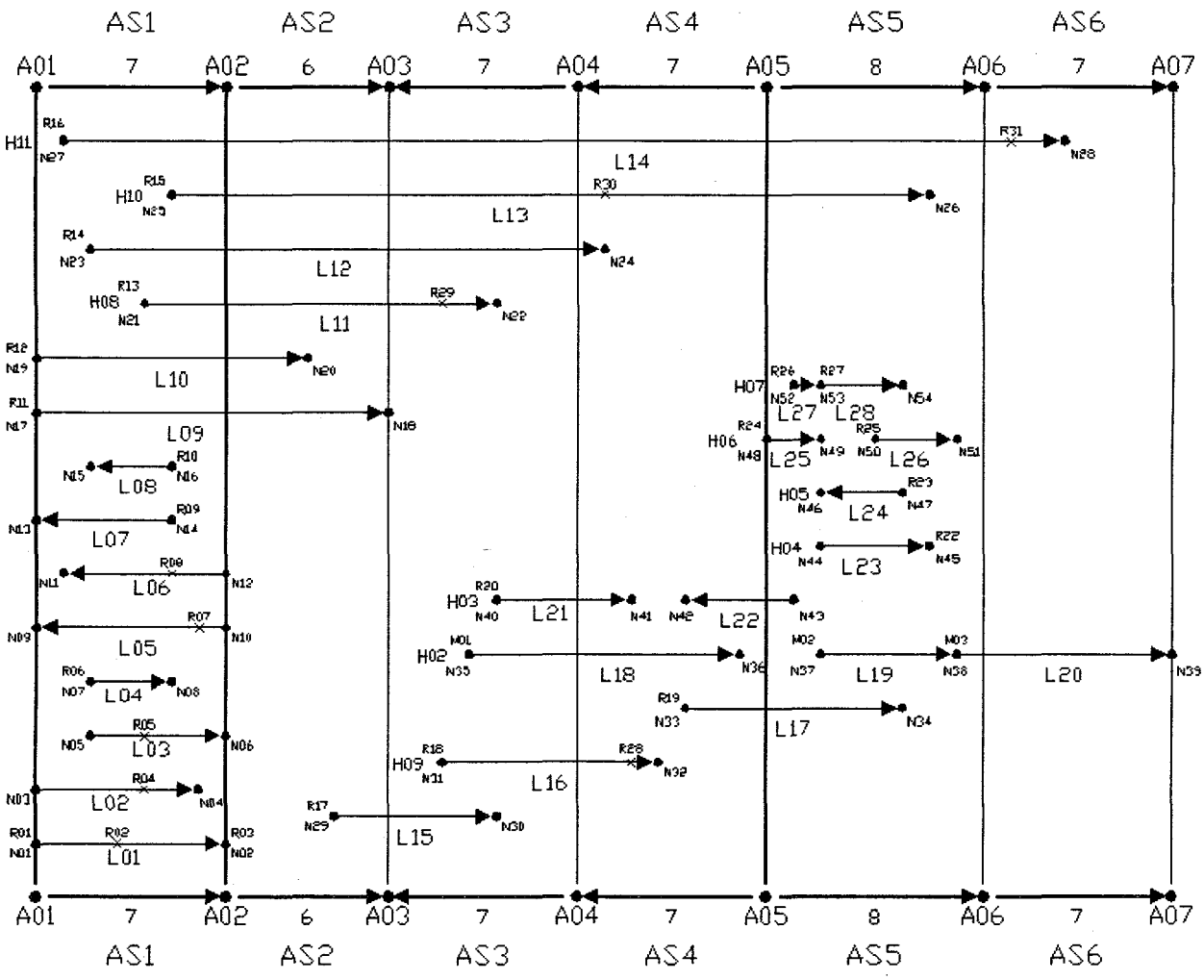
Not applicable

### 3.6.3 Site Adaptation

Upon installation of the product, the user must run the Delphi Database Engine Configuration from the Delphi group in Windows. Under the Aliases tab, create a new alias 'LRMtest' with the pathname to the directory containing the database tables shipped with the LRM executable.

## 4. APPENDIX A: TEST CASES

The test cases correspond to the test network graphed below.


AS1
AS2
AS3
AS4
AS5
AS6
A01
7
A02
6
A03
7
A04
7
A05
8
A06
7
A07
H11
H10
H08
H07
H06
H05
H04
H03
H02
H09
L01
L02
L03
L04
L05
L06
L07
L08
L09
L10
L11
L12
L13
L14
L15
L16
L17
L18
L19
L20
L21
L22
L23
L24
L25
L26
L27
L28

The following cases test the correct conversion of Reference Post address to Datum Address. The second Reference Post output in a Post+offset cell is the observed Reference Post output, mapped to the first traversal looked up in the reverse algorithm.

| | Input | | | Expected output | | |
|---|---|---|---|---|---|---|
| ID | Sys | Trav | Post+offset | Datum | Mpt | Explanation of test case |
| | S01 | H01 | R01+0.00 | AS1+0.00 | 0.00 | Nominal, beginning of link, same dir |
| | S01 | H01 | R02+0.00 | AS1+0.03 | 0.03 | Middle of link |
| | S01 | H01 | R03+0.00 | AS1+0.07 | 0.07 | End of link |
| | S01 | H01 | R04+0.00 | AS1+0.04 | 0.04 | Same dir, link shorter than section at head |
| | S01 | H01 | R05+0.00<br>R02+0.01 | AS1+0.04 | 0.04 | Same dir, link shorter at tail |
| | S01 | H01 | R06+0.00<br>R01+0.02 | AS1+0.02 | 0.02 | Same dir, link entirely in section |
| | S01 | H01 | R07+0.00<br>R02+0.03 | AS1+0.06 | 0.06 | Link opposite dir, full span over section |
| | S01 | H01 | R08+0.00<br>R02+0.02 | AS1+0.05 | 0.05 | Opposite dir, link shorter at head |
| | S01 | H01 | R09+0.00<br>R02+0.02 | AS1+0.05 | 0.05 | Opposite dir, link shorter at tail |
| | S01 | H01 | R10+0.00<br>R02+0.02 | AS1+0.05 | 0.05 | Opposite dir, link entirely in section |
| | S01 | H01 | R11+0.00<br>R01+0.00 | AS1+0.00 | 0.00 | Nominal, beginning of link, same dir |
| | S01 | H01 | R11+0.13 | AS2+0.06 | N/A | End of second section, same dir |
| | S01 | H01 | R12+0.00<br>R01+0.00 | AS1+0.00 | 0.00 | Nominal, beginning of link, same dir |
| | S01 | H01 | R14+0.00<br>R01+0.02 | AS1+0.02 | 0.02 | Nominal, beginning of link, same dir |
| | S01 | H01 | R17+0.00<br>R11+0.11 | AS2+0.04 | N/A | Nominal, beginning of link, same dir |
| | S01 | H01 | R19+0.00<br>R14+0.22 | AS4+0.03 | N/A | Nominal, beginning of link, opposite dir |
| | S01 | H08 | R13+0.00 | AS1+0.04 | 0.00 | Nominal, beginning of link, same dir |
| | S01 | H08 | R13+0.04 | AS2+0.01 | 0.04 | Crossing head to tail section join |
| | S01 | H01 | R17+0.04<br>R11+0.15 | AS3+0.05 | N/A | Crossing head to head section join |
| | S01 | H09 | R18+0.06 | AS4+0.06 | 0.06 | Crossing tail to head section join |
| | S01 | H01 | R19+0.04 | AS5+0.01 | N/A | Crossing tail to tail section join |
| | S01 | H01 | R02+0.04<br>R03+0.00 | AS1+0.07 | 0.07 | Nominal, same dir, plus offset |
| | S01 | H01 | R07+0.04<br>R01+0.02 | AS1+0.02 | 0.02 | Nominal, opposite dir, plus offset |
| | S01 | H01 | R08+0.04<br>R01+0.01 | AS1+0.01 | 0.01 | Opposite dir, to end of link |
| | S01 | H09 | R18+0.00 | AS3+0.05 | 0.00 | Nominal, beginning of link, opposite dir |

The following cases test the correct conversion of Milepoint Address to Datum Address.

| | Input | | | Expected output | | |
|---|---|---|---|---|---|---|
| ID | Sys | Trav | Mpt | Datum | Post+offset | Explanation of test case |
| | S01 | H02 | 0.00 | AS3+0.04 | N/A | Nominal |
| | S01 | H02 | 0.04 | AS3+0.00 | N/A | Offset to section boundary |
| | S01 | H02 | 0.10 | AS4+0.01 | N/A | End of section |
| | S01 | H02 | 0.11 | AS5+0.03 | N/A | Over discontinuity between links |
| | S01 | H02 | 0.15 | AS5+0.07 | N/A | End of second link |
| | S01 | H02 | 0.16 | AS5+0.08 | N/A | Third link |
| | S01 | H02 | 0.23 | AS6+0.07 | N/A | End of third link |

The following cases test the correct conversion of Reference Post address to Datum address and then to Milepoint address.

| | Input | | | Expected output | | |
|---|---|---|---|---|---|---|
| ID | Sys | Trav | Post+offset | Datum | Mpt | Explanation of test case |
| | S01 | H11 | R16+0.00 | AS1+0.01 | 0.00 | Nominal, beginning of link, same dir |
| | S01 | H11 | R16+0.06 | AS1+0.07 | 0.06 | Section boundary |
| | S01 | H11 | R16+0.07 | AS2+0.01 | 0.07 | Crossing head to tail section join |
| | S01 | H11 | R16+0.14 | AS3+0.05 | 0.14 | Third section, head to head join |
| | S01 | H11 | R16+0.21 | AS4+0.05 | 0.21 | Fourth section, tail to head join |
| | S01 | H11 | R16+0.29 | AS5+0.03 | 0.29 | Fifth section, tail to tail join |
| | S01 | H11 | R16+0.37 R31+0.02 | AS6+0.03 | 0.37 | End, sixth section, head to tail join |

The following cases test the correct conversion of traversal Milepoint address to Reference Post address. The Datum Address is listed for completeness.

| | Input | | | Expected output | | |
|---|---|---|---|---|---|---|
| ID | Sys | Trav | Mpt | Post+offset | Datum | Explanation of test case |
| | S01 | H01 | 0.00 | R01+0.00 | AS1+0.00 | Exactly on refpost |
| | S01 | H01 | 0.02 | R01+0.02 | AS1+0.02 | Refpost plus offset |
| | S01 | H01 | 0.03 | R02+0.00 | AS1+0.03 | Refpost nearness math. Not R01+0.03. |
| | S01 | H04 | 0.03 | R22-0.01 | AS5+0.05 | Refpost occurs at end of link, neg offset |
| | S01 | H04 | 0.04 | R22+0.00 | AS5+0.06 | Refpost occurs at end of link, zero offset |
| | S01 | H05 | 0.00 | R23+0.00 | AS5+0.05 | Link goes opposite direction of section. |
| | S01 | H05 | 0.01 | R23+0.01 | AS5+0.04 | Link goes opposite section, offset math |
| | S01 | H07 | 0.04 | R27+0.01 | AS5+0.03 | Milepoint spans two links |
| | S01 | H08 | 0.10 | R13+0.10 | AS3+0.06 | Link spans three sections |
| | S01 | H09 | 0.00 | R18+0.00 | AS3+0.05 | Link goes opposite direction of section |
| | S01 | H09 | 0.02 | R18+0.02 | AS3+0.03 | Link goes opposite section, offset math |
| | S01 | H09 | 0.07 | R28+0.00 | AS4+0.05 | Link spans two opposite dir sections |
| | S01 | H10 | 0.04 | R15+0.04 | AS2+0.02 | Link spans two same direction sections |
| | S01 | H10 | 0.11 | R15+0.11 | AS3+0.04 | Link spans three sections |
| | S01 | H10 | 0.23 | R30+0.07 | AS5+0.01 | Link spans five sections |
| | S01 | H11 | 0.35 | R31+0.00 | AS6+0.01 | Link spans six sections, zero offset |

Intentionally Left Blank

DISTRIBUTION:

2 University of New Mexico
Alliance for Transportation Research
ATTN: David R. Fletcher
1001 University Blvd. SE, Suite 103
Albuquerque, NM 87106-4342

2 New Mexico State Highway and Transportation Dept.
ATTN: Thomas Henderson
1120 Cerrillos Road, SB-3
Santa Fe, New Mexico 87504-1149

2 University of Wisconsin-Madison
Department of Civil and Environmental Engineering
ATTN: Alan Vonderohe, T. Hepworth
1208 Engineering Hall
Madison, WI 53706

1 Oak Ridge National Laboratory
ATTN: Stephen R. Gordon
P.O. Box 2008
Bldg. 4500N, MS 6270
Oak Ridge, TN 37831

1 U.S. Department of Transportation
Bureau of Transportation Statistics
ATTN: Bruce Spear
400 7th Street SW
Washington, DC 20590

| | | | |
|---|---|---|---|
| 1 | MS | 0188 | Chuck Meyers, 4523 |
| 5 | | 0775 | Juan Espinoza Jr., 6533 |
| 1 | | 0775 | R. D. Mackoy, 6533 |
| 1 | | 1125 | Wendy A. Amai, 5516 |
| 1 | | 1138 | Sharon K. Chapa, 6533 |
| 1 | | 1138 | Craig Dean, 6533 |
| 1 | | 1138 | Hillary Armstrong, 6533 |
| 1 | | 9018 | Central Technical Files, 8940-2 |
| 5 | | 0899 | Technical Library, 4916 |
| 2 | | 0619 | Review & Approval Desk, 12690<br>For DOE/OSTI |