

5-1291-01-P2

INTERACTIVE GRAPHICS INTERSECTION DESIGN SYSTEM (IGIDS) USERS MANUAL: UPDATE TO APPENDIX C

Authors:

Thomas W. Rioux

Robert F. Inman

Randy B. Machemehl

Project 5-1291-01: Interactive Graphics Intersection Design System (IGIDS)

OCTOBER 2001

Performing Organization: Center for Transportation Research The University of Texas at Austin 3208 Red River, Suite 200 Austin, Texas 78705-2650		Sponsoring Organization: Texas Department of Transportation Research and Technology Implementation Office P.O. Box 5080 Austin, Texas 78763-5080	
Project conducted in cooperation with the Federal Highway Commission and the Texas Department of Transportation.			
Description: User manual for the IGIDS system.	Keywords: Interactive, graphics, design, IGIDS, user's manual, MicroStation, geometric design, environmental, hydraulics, right-of-way		No. of Pages: 160

Abstract

The Interactive Graphics Intersection Design System (IGIDS) is a computer software program developed for TxDOT that assists engineers in the analysis and design of individual, at-grade intersections. IGIDS was initially developed under several RMC 3 - Geometric Design, Environmental, Hydraulics, and Right-of-Way projects. IGIDS is a MicroStation application which contains (1) geometric, traffic data, and signalization definition tools; (2) built-in analysis tools for vehicle turning templates; horizontal sight distance checking for stop sign, yield sign, and no control; and "Highway Capacity Manual" (HCM), Chapter 9 procedures; and (3) interfaces to external analysis programs for the TEXAS Model for Intersection Traffic (TEXAS), the TxDOT Automated Plan Preparation System (APP), and the Signal Operations and Analysis Package (SOAP). IGIDS can operate in English or metric units. IGIDS was recently enhanced by the addition of vertical sight distance checking within the horizontal sight triangle, the inclusion of pavement striping definition and tabulation, and the development of training materials.

Most TxDOT engineers now use GEOPAK for the geometric design of intersections. TxDOT Traffic Operations personnel generally have access to the GEOPAK design files for an intersection. IGIDS was enhanced to directly read the intersection geometry from an existing GEOPAK design file. The addition of this feature makes IGIDS much easier to use and reduce duplicated effort. In addition, most TxDOT engineers now prefer to use PASSER II-90 for signal timing optimization rather than using SOAP. IGIDS was enhanced by the addition of a PASSER II-90 interface. The addition of this feature increases the engineers' ability to analyze and design an intersection for optimum operation. Since the last release of IGIDS, the HCM Chapter 9 procedures have been updated. IGIDS was modified to use the 1998 version of the HCM Chapter 9 procedures. Training modules for these enhancements were developed and added to the training course previously developed for IGIDS. In addition, each training module was updated to reflect recent experience using web-based training. Finally, a test training session was held to evaluate the training materials and procedures.

This document updates Research Report 1308-1F Interactive Graphics Intersection Design User's Manual - Appendix C, Center for Transportation Research, The University of Texas at Austin, December 1994.

APPENDIX C

IGIDS Command Descriptions

This appendix contains the IGIDS Diagram Notes, the IGIDS Command Overview, and the IGIDS Command Descriptions. The IGIDS Command Descriptions are in alphabetical order.

IGIDS Diagram Notes

Square cornered boxes enclose a prompt to be presented to the user. Each prompt requires a user input for processing to continue.

Round cornered boxes are for reporting actions that are taken by IGIDS.

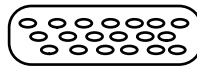
Diamond shaped boxes enclose a question. Some of the corners are labeled with possible answers to the question. The answer determines the path to be taken from this box. Most of these questions have a "yes" or "no" answer.

There are 4 types of user input:

- (1) Reset - Executed by pressing the mouse reset button. Represented by this symbol:



- (2) Keyin - A sequence of key presses that are ended by a pressing the return key. Represented by this symbol:



- (3) DataPt - Executed by using the mouse to position the cursor at the desired coordinates and then pressing the mouse data button. Represented by this symbol:



- (4) Reenter - An IGIDS command to request that IGIDS re-prompt for the most recently keyed-in data. Represented by this symbol:



Each line leaving a square cornered box is labeled with the authorizing event. These labels are placed upon the lines.

Each line entering an Identify Object box is labeled to indicate the point of entry into the Identify Object operation. These labels are adjacent to the lines.

Diagrams that are for more than one command have underscores (_____) substituted for words that are specific to a command. An example is the **MOVE - TEXT** diagram. This is used for both **Primitive Command: MOVE - TEXT ON SEG** and **Primitive Command: MOVE - TEXT ON ALT**. In the Identify Object block for this diagram, the underscore may be replaced by either the word Alternative or the word Seg, as appropriate.

identify Text on _____
DataPt/Reset: identify Text on _____/end command
DataPt/Reset: accept/reidentify

Underscores are also used to replace parts of prompts that may vary, such as the Leg number when prompting for the volume on a Leg.

IGIDS Command Overview

IGIDS Commands

An IGIDS command is a request from the user for IGIDS to take some action. IGIDS commands are initiated by a Click on an IGIDS menu item. There are three types of IGIDS commands as documented in this Appendix:

- (1) **Primitive Command:** an IGIDS command that initiates an action that requires user interaction. When issued during the execution of another IGIDS Command or a MicroStation Command, it cancels any active IGIDS or MicroStation command.
- (2) **Temporary Command:** an IGIDS command that initiates an action that requires user interaction. When issued during the execution of another IGIDS Command, it temporarily suspends the command in progress. When the temporary command is ended, the suspended command continues from the point where it was suspended.
- (3) **Transient Command:** an IGIDS command that initiates an action that requires no user interaction. When issued during the execution of another IGIDS Command, it does not end the command in progress.

Identify Object subcommand

An operation that is common to many commands is the identification of an existing IGIDS object for processing. When in the **Verb-Noun** mode, the user is prompted to identify the Object of choice by placing a data point (DataPt) near the Object. An Object near the DataPt is then hilited. When in the **Noun-Verb** mode, the selected Object of the appropriate type is hilited automatically. Next, in either mode, the user is prompted to confirm with a DataPt or deny with a Reset that the hilited Object is the correct one. If the hilited Object is not the Object of choice, the user will be prompted to identify another Object with a new DataPt.

If there is only one object of the specified type, it will be assumed to be the object of choice and will automatically become the identified object. The diagram of this operation is shown below in Figure 1(a). When a part of the diagram for an IGIDS command, it will be represented in the simplified form as shown in Figure 1(b).

As Figure 1(b) shows, there are 3 points of entry into and 3 points of exit from the identification diagram. All of these points may not be appropriate for some commands. Only the entry and exit points that are needed will be used in diagrams for specific commands.

The identify Object block in Figure 1(c) is used in **Primitive Command: ROTATE - LEG**. The exit by Reset and entry by DataPt are not needed for this command so are not shown in the diagram. As is typical, the placement of the entry and exit points has been adjusted to suit the particular diagram.

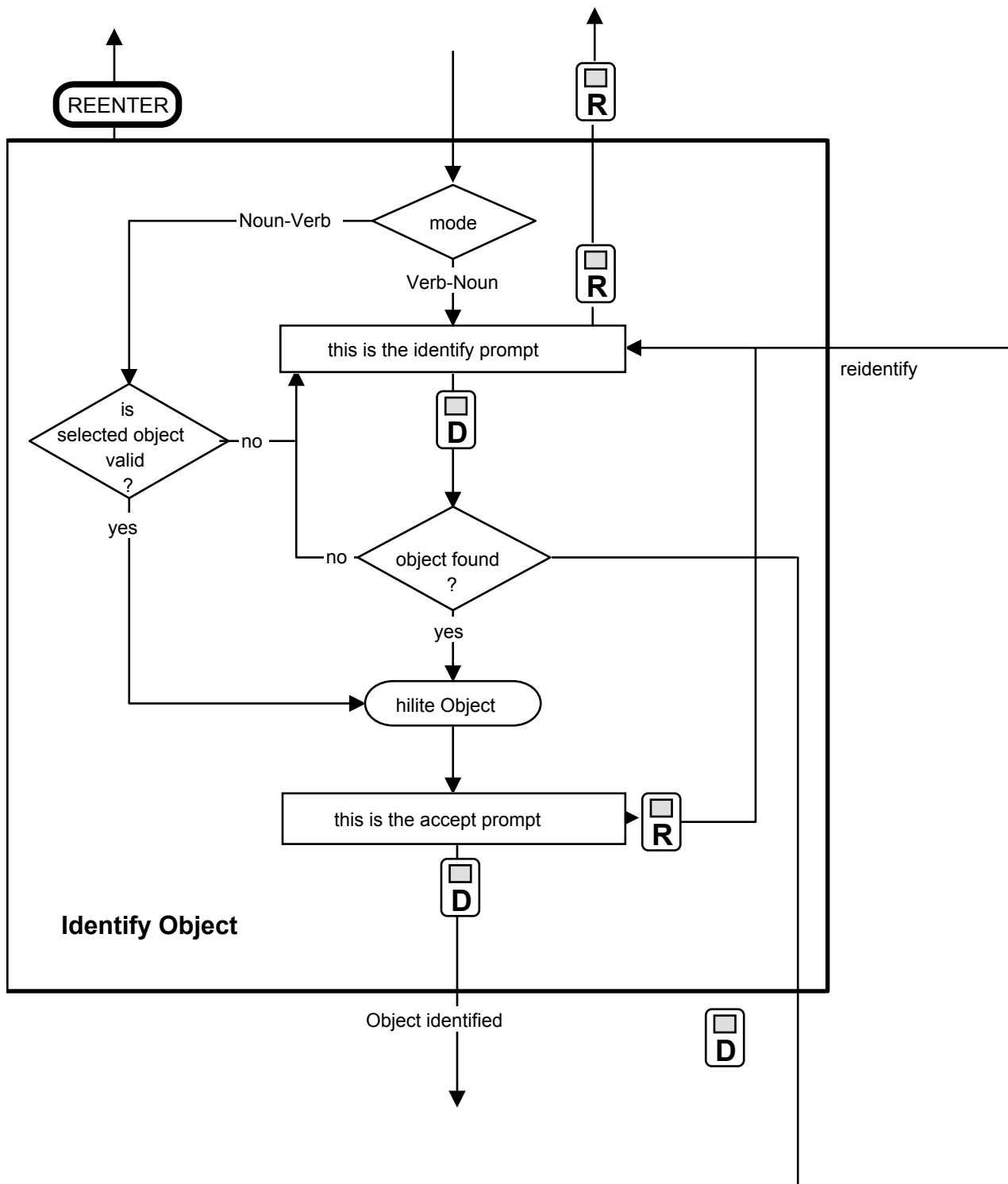


Figure 1(a) Identify object.

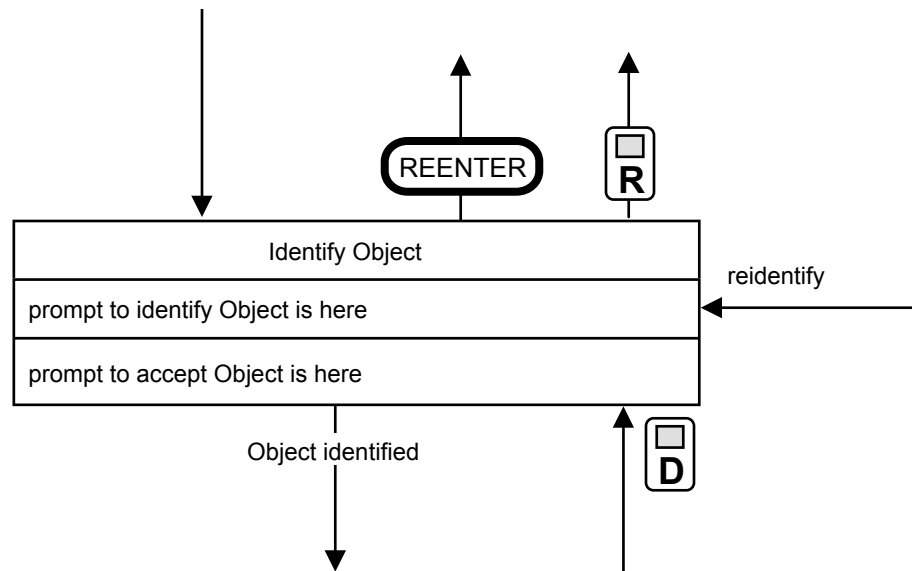


Figure 1(b) Identify object as shown in diagrams.

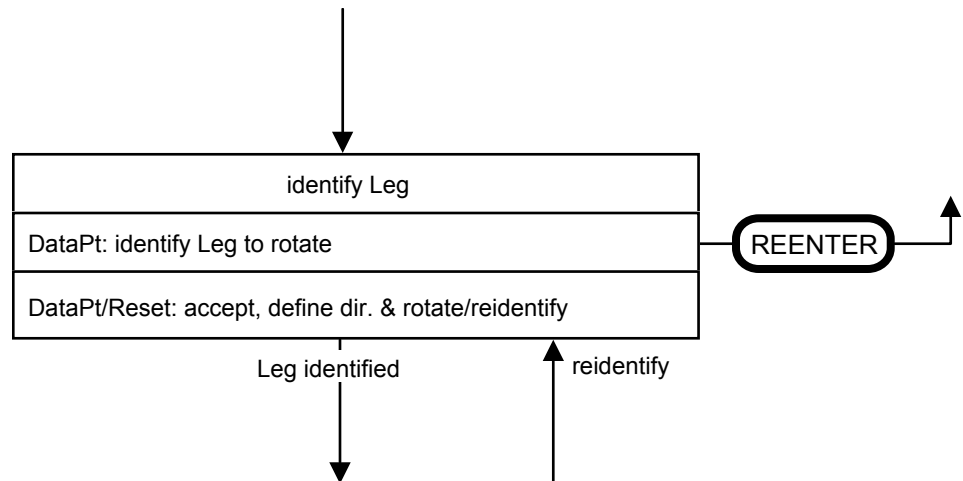


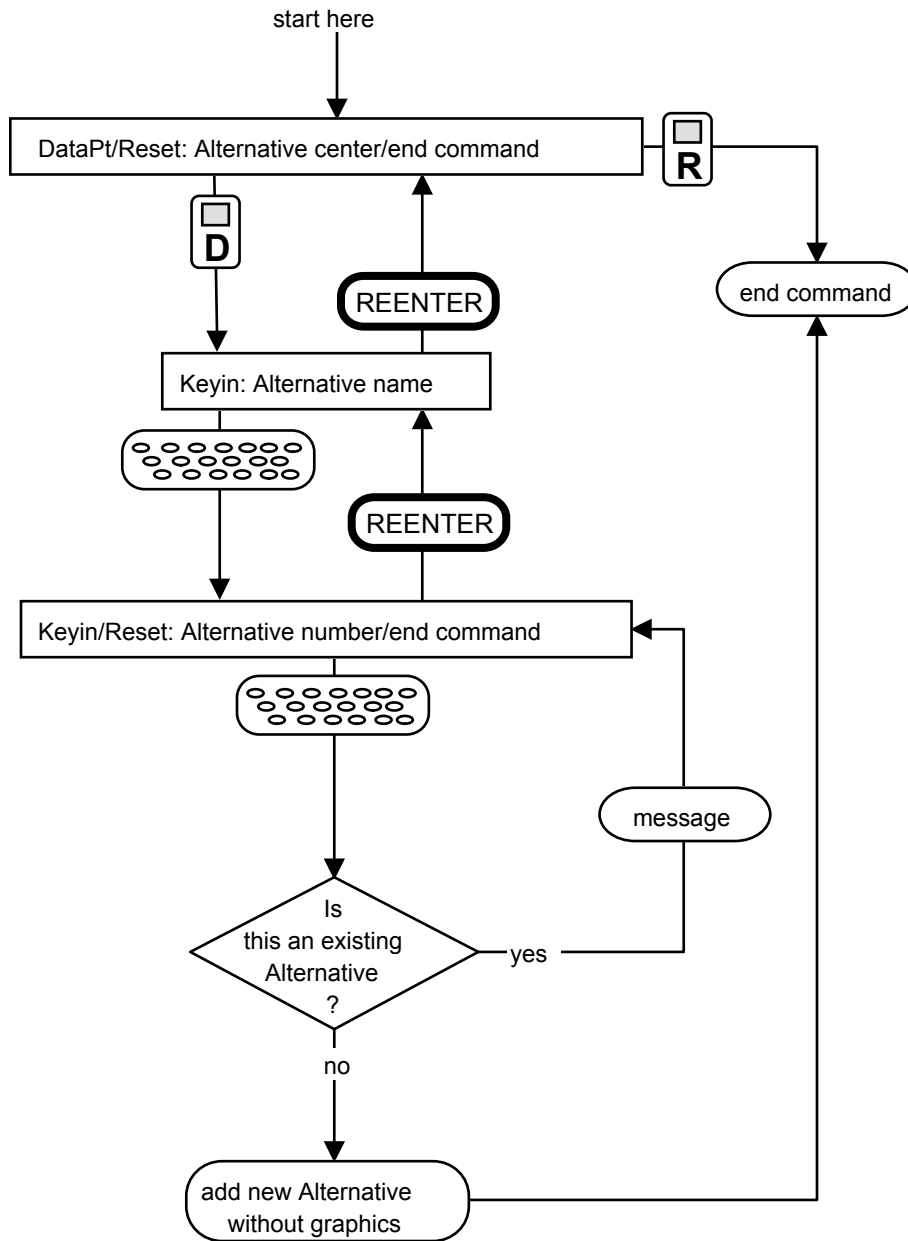
Figure 1(c) Identify object as in the **ROTATE-LEG** diagram.

IGIDS Command Descriptions

The IGIDS Command Descriptions are in alphabetical order. The major groupings are:

- (1) **ADD,**
- (2) **COPY,**
- (3) **DELETE,**
- (4) **END IGIDS,**
- (5) **HILITE,**
- (6) **LOAD FROM,**
- (7) **MODIFY,**
- (8) **MOVE,**
- (9) **No,**
- (10) **Noun - Verb and Verb - Noun,**
- (11) **Reenter Data,**
- (12) **ROTATE,**
- (13) **SAVE TO**
- (14) **SELECT,**
- (15) **SHOW INFO,**
- (16) **Sta/Offset,**
- (17) **TOOLS,**
- (18) **VIEW,**
- (19) **Yes, and**
- (20) **[default].**

Primitive Command: ADD - ALTERNATIVE



Alternative center coordinates, name and ID number are specified. There is no graphical evidence that the Alternative was added.

Primitive Command: ADD - LANE INBND - BY KEY-IN

identify Leg for adding Lanes

keyin: not allowed
datapt: select object
reset: end command
reenter: not allowed

Keyin/Reset: no. new inb. Lanes[1]/reidentify

keyin: integer; <1 or > 6 then error message then prompt again
datapt: not allowed
reset: backup to identify Leg for adding Lanes
reenter: not allowed

if no. new inb. Lanes > 1 then Keyin: number of left turn bays [0]

keyin: integer; if < 0 or > no. new inb. Lanes then error message then prompt again
datapt: not allowed
reset: not allowed
reenter: backup to Keyin/Reset: no. new inb. Lanes[1]/reidentify

if no. new inb. Lanes > number of left turn bays then Keyin: number of right turn bays [0]

keyin: integer; if < 0 or > no. new inb. Lanes - number of left turn bays then error message then prompt again
datapt: not allowed
reset: not allowed
reenter: backup to Keyin/Reset: no. new inb. Lanes[1]/reidentify

Keyin/DataPt: setback to beginning of Lane

keyin: double; if < 0 then error message then prompt again; default distance from CL = 0
datapt: get station and offset of datapt; if station not within centerline's beginning and ending station then error message and prompt again; if station number invalid then error message and prompt again; setback = fabs (centerline station - station of datapt); if < 0 then error message then prompt again; default distance from CL = offset of datapt
reset: not allowed
reenter: backup to (a) Keyin: number of right turn bays, (b) Keyin: number of left turn bays, or (c) Keyin/Reset: no. new inb. Lanes[1]/reidentify as appropriate

Keyin/DataPt: distance from CL [default distance from CL]

keyin: double
datapt: get station and offset of datapt; if station not within centerline's beginning and ending station then error message and prompt again; if station number invalid then error message and prompt again; distance from CL = offset of datapt properly signed
reset: not allowed
reenter: backup to Keyin/DataPt: setback to beginning of Lane

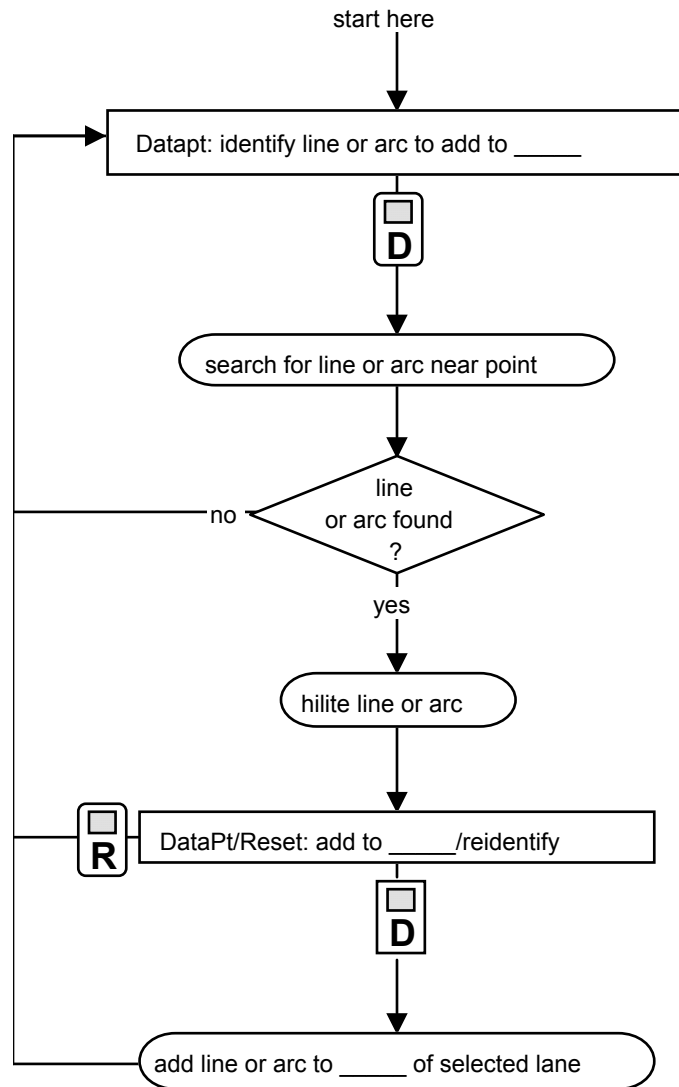
Keyin/DataPt: width of one Lane [default lane width]/all Lanes

keyin: double; if < minimum lane width then error message then prompt again; if > maximum lane width then error message and prompt again; default lane length = centerline length - setback to beginning of Lane
datapt: get station and offset of datapt; if station not within centerline's beginning and ending station then error message and prompt again; if station number invalid then error message and prompt again; width of one Lane = fabs (distance from CL - fabs (offset of datapt)) / number of lanes; if < minimum lane width then error message then prompt again; if > maximum lane width then error message and prompt again; default lane length = station of datapt - setback to beginning of Lane
reset: not allowed
reenter: backup to Keyin/DataPt: distance from CL

Keyin/DataPt: Lane length [default lane length]

keyin: double; if < minimum lane length then error message then prompt again
datapt: get station and offset of datapt; if station not within centerline's beginning and ending station then error message and prompt again; if station number invalid then error message and prompt again; Lane length = fabs (centerline beginning station - station of datapt) - setback to beginning of Lane; if < minimum lane length then error message then prompt again
reset: not allowed
reenter: backup to Keyin/DataPt: width of one Lane
add inbound lane(s)

Primitive Command: ADD - LANE INBND - SCRATCH LVL



Use this diagram for the 3 commands that follow:

Primitive Command: ADD - LANE INBND - SCRATCH LVL - INNER EDGE

Add a duplicate of an existing line or arc to the inner edge of the selected Lane. The existing line or arc cannot be an IGIDS Object, but must be on the scratch level or in an attached reference file.

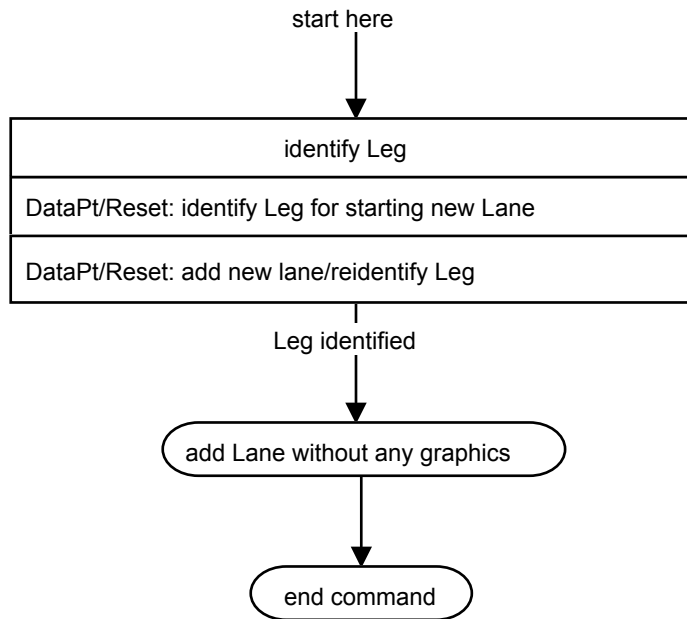
Primitive Command: ADD - LANE INBND - SCRATCH LVL - OUTER EDGE

Add a duplicate of an existing line or arc to the outer edge of the selected Lane. The existing line or arc cannot be an IGIDS Object, but must be on the scratch level or in an attached reference file.

Primitive Command: ADD - LANE INBND - SCRATCH LVL - STOP LINE

Add a duplicate of an existing line or arc to the stop line of the selected Lane. The existing line or arc cannot be an IGIDS Object, but must be on the scratch level or in an attached reference file.

Primitive Command: ADD - LANE INBND - SCRATCH LVL - START LANE



Add a new Inbound Lane to the selected Leg. The new Lane is made the selected Lane. There is no graphical evidence that the Lane has been added.

Primitive Command: ADD - LANE OUTBND - BY KEY-IN

identify Leg for adding Lanes

keyin: not allowed
datapt: select object
reset: end command
reenter: not allowed

Keyin/Reset: no. new outb. Lanes [1]/reidentify

keyin: integer; <1 or > 6 then error message then prompt again
datapt: not allowed
reset: backup to identify Leg for adding Lanes
reenter: not allowed

Keyin: number of right turn bays [0]

keyin: integer; if < 0 or > no. new outb. Lanes then error message then prompt again
datapt: not allowed
reset: not allowed
reenter: backup to Keyin/Reset: no. new outb. Lanes [1]/reidentify

Keyin/DataPt: setback to beginning of Lane

keyin: double; if < 0 then error message then prompt again; default distance from CL = 0
datapt: get station and offset of datapt; if station not within centerline's beginning and ending station then error message and prompt again; if station number invalid then error message and prompt again; setback = fabs (centerline station - station of datapt); if < 0 then error message then prompt again; default distance from CL = offset of datapt
reset: not allowed
reenter: backup to Keyin: number of right turn bays

Keyin/DataPt: distance from CL [default distance from CL]

keyin: double
datapt: get station and offset of datapt; if station not within centerline's beginning and ending station then error message and prompt again; if station number invalid then error message and prompt again; distance from CL = offset of datapt properly signed
reset: not allowed
reenter: backup to Keyin/DataPt: setback to beginning of Lane

Keyin/DataPt: width of one Lane [default lane width]/all Lanes

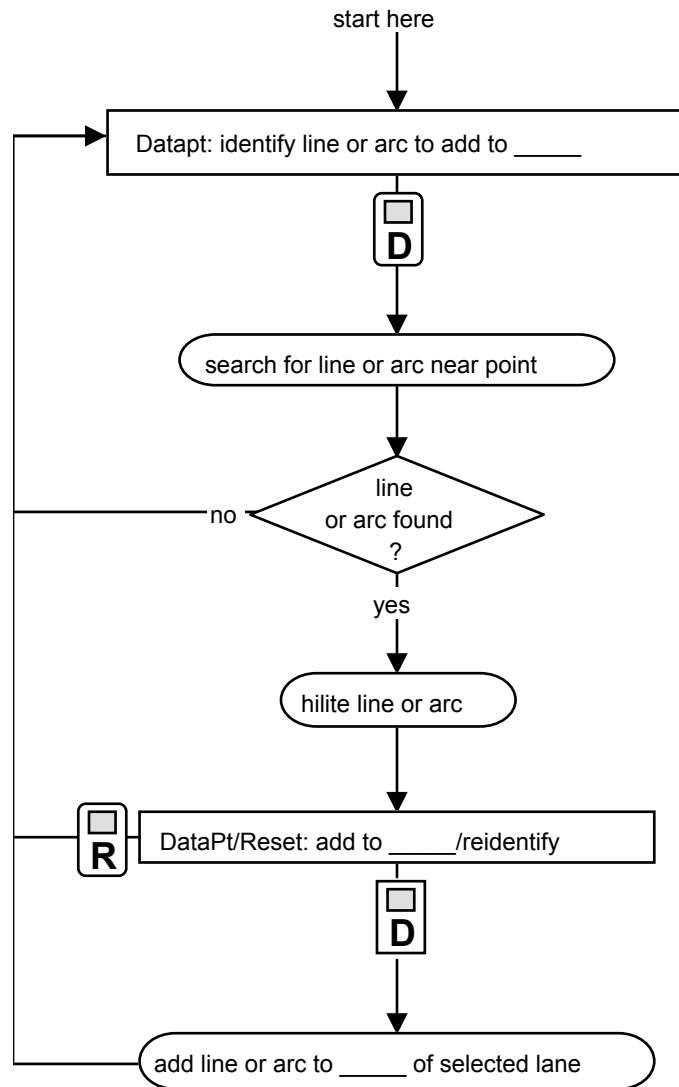
keyin: double; if < minimum lane width then error message then prompt again; if > maximum lane width then error message and prompt again; default lane length = centerline length - setback to beginning of Lane
datapt: get station and offset of datapt; if station not within centerline's beginning and ending station then error message and prompt again; if station number invalid then error message and prompt again; width of one Lane = fabs (distance from CL - fabs (offset of datapt)) / number of lanes; if < minimum lane width then error message then prompt again; if > maximum lane width then error message and prompt again; default lane length = station of datapt - setback to beginning of Lane
reset: not allowed
reenter: backup to Keyin/DataPt: distance from CL

Keyin/DataPt: Lane length [default lane length]

keyin: double; if < minimum lane length then error message then prompt again
datapt: get station and offset of datapt; if station not within centerline's beginning and ending station then error message and prompt again; if station number invalid then error message and prompt again; Lane length = fabs (centerline beginning station - station of datapt) - setback to beginning of Lane; if < minimum lane length then error message then prompt again
reset: not allowed

reenter: backup to Keyin/DataPt: width of one Lane
add outbound lane(s)

Primitive Command: ADD - LANE OUTBND - SCRATCH LVL



Use this diagram for the 3 commands that follow:

Primitive Command: ADD - LANE OUTBND - SCRATCH LVL - INNER EDGE

Add a duplicate of an existing line or arc to the inner edge of the selected Lane. The existing line or arc cannot be an IGIDS Object, but must be on the scratch level or in an attached reference file.

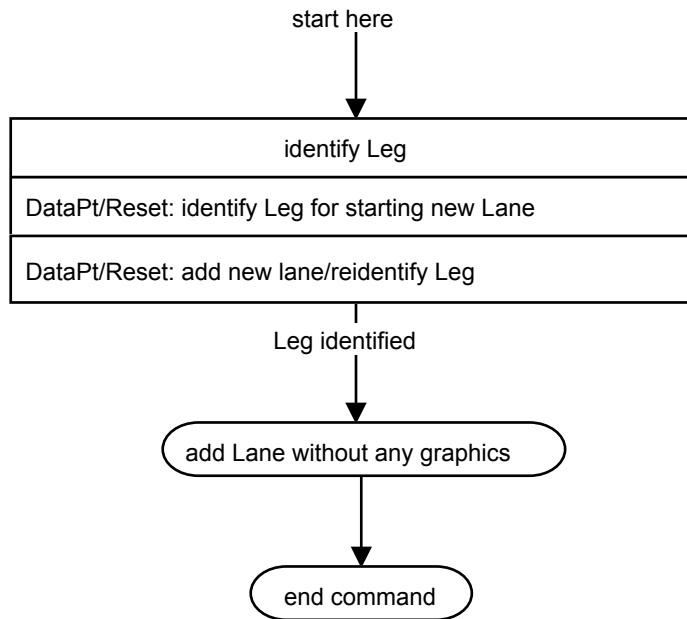
Primitive Command: ADD - LANE OUTBND - SCRATCH LVL - OUTER EDGE

Add a duplicate of an existing line or arc to the outer edge of the selected Lane. The existing line or arc cannot be an IGIDS Object, but must be on the scratch level or in an attached reference file.

Primitive Command: ADD - LANE OUTBND - SCRATCH LVL - STOP LINE

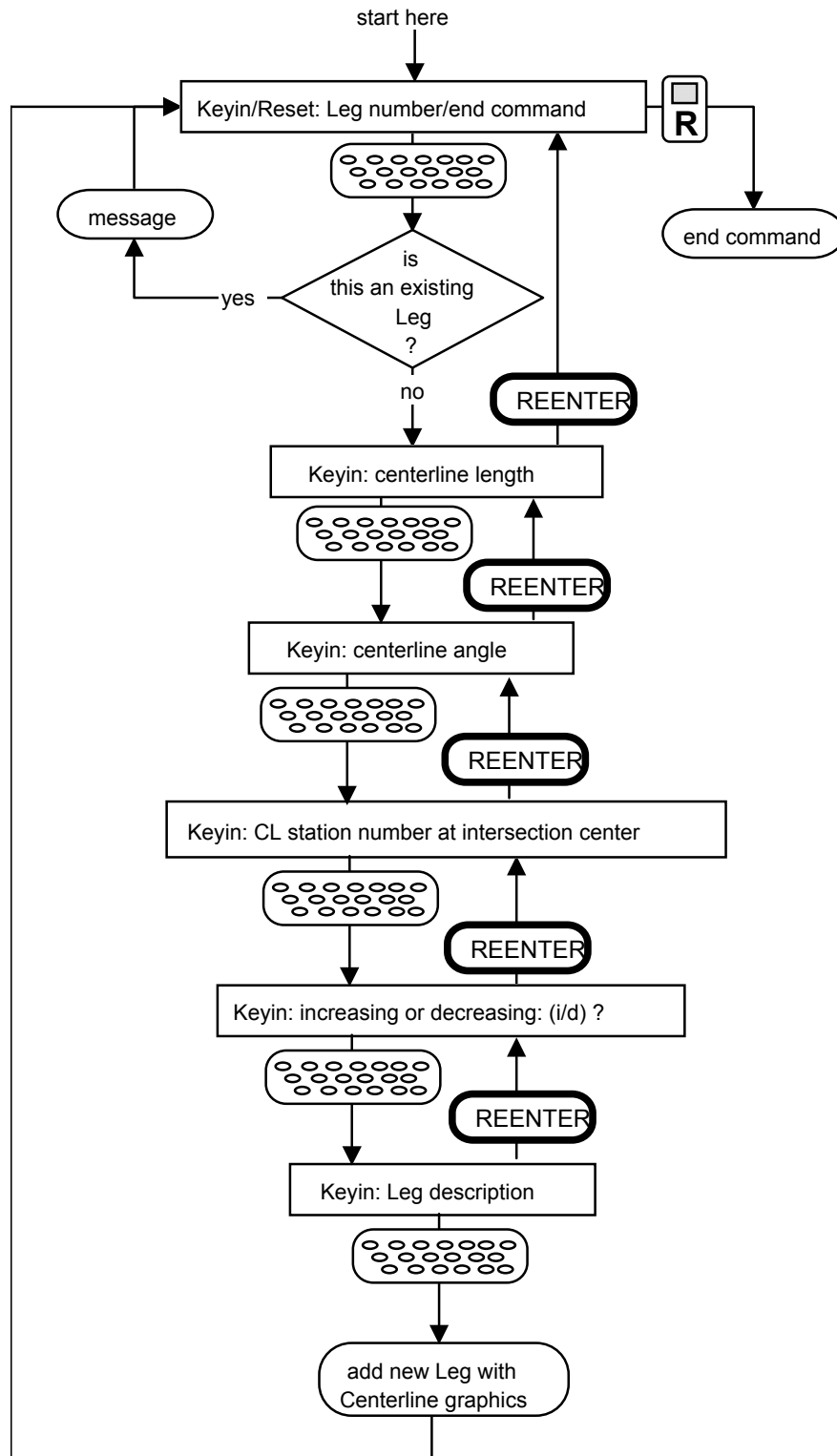
Add a duplicate of an existing line or arc to the stop line of the selected Lane. The existing line or arc cannot be an IGIDS Object, but must be on the scratch level or in an attached reference file.

Primitive Command: ADD - LANE OUTBND - SCRATCH LVL - START LANE



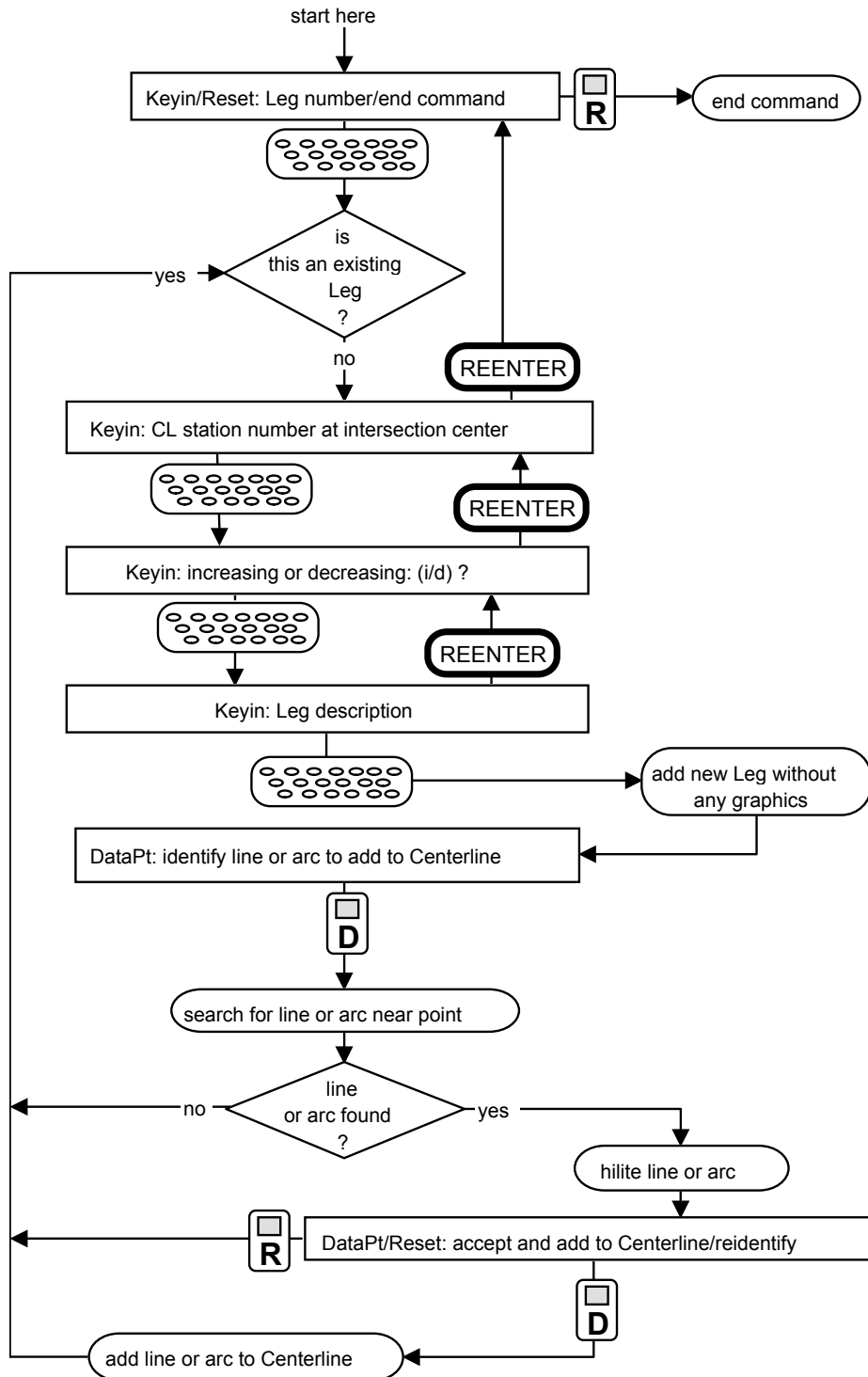
Add a new Outbound Lane to the selected Leg. The new Lane is made the selected Lane. There is no graphical evidence that the Lane has been added.

Primitive Command: ADD - LEG CNTRLN - BY KEY-IN



Add a new Leg to the selected Alternative. Leg ID number, centerline length, centerline angle, station number at center of intersection and direction of stationing and Leg description must be specified. The centerline will be one straight Seg and will start at the center of the Alternative.

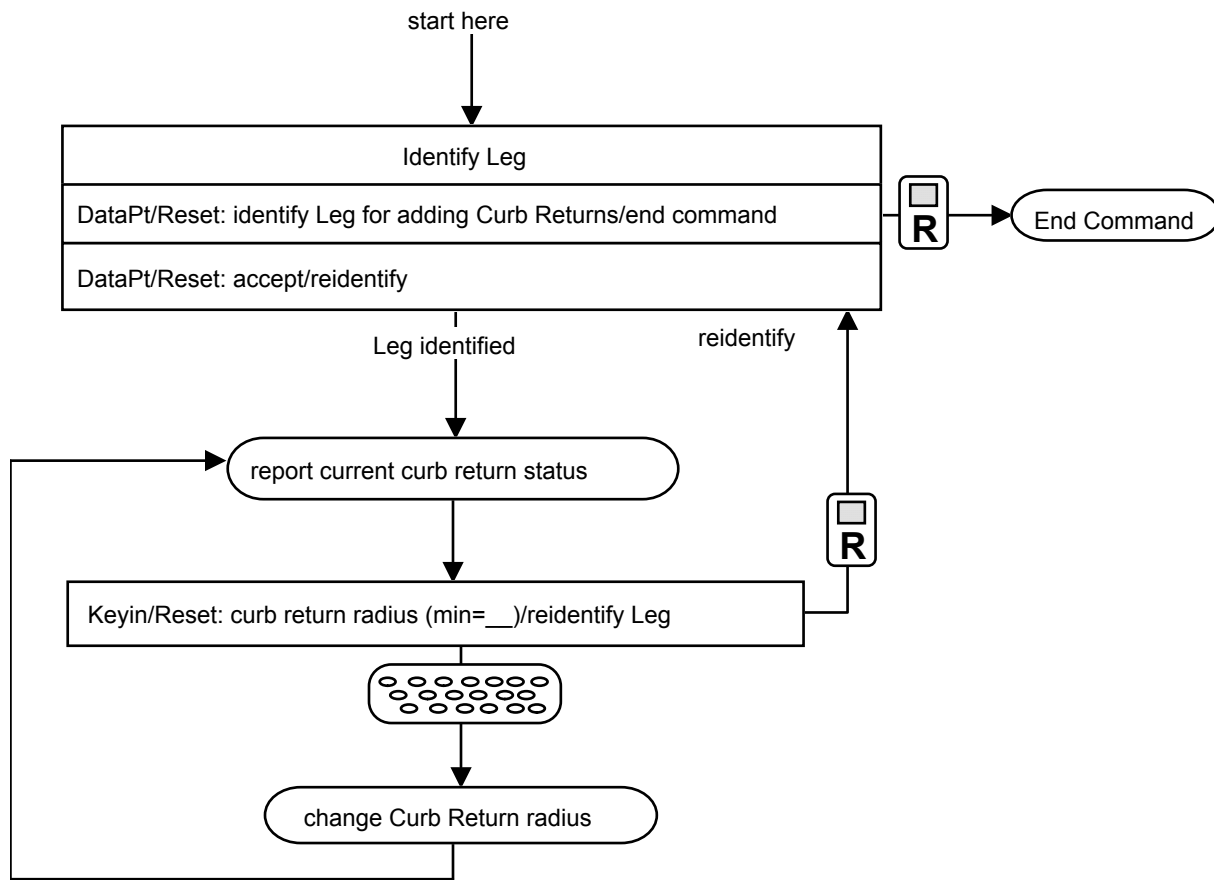
Primitive Command: ADD - LEG CNTRLN - SCRATCH LVL



For the selected Alternative, add a new Leg and/or Centerline Segs. Leg ID number is specified. If the leg doesn't exist, add a new Leg. For the new Leg, station number at center of intersection and direction of stationing and Leg description must be specified. Duplicates of existing lines or arcs may be added to

the new or an existing Leg Centerline. The existing lines or arcs cannot be IGIDS Objects, but must be on the scratch level or in an attached reference file.

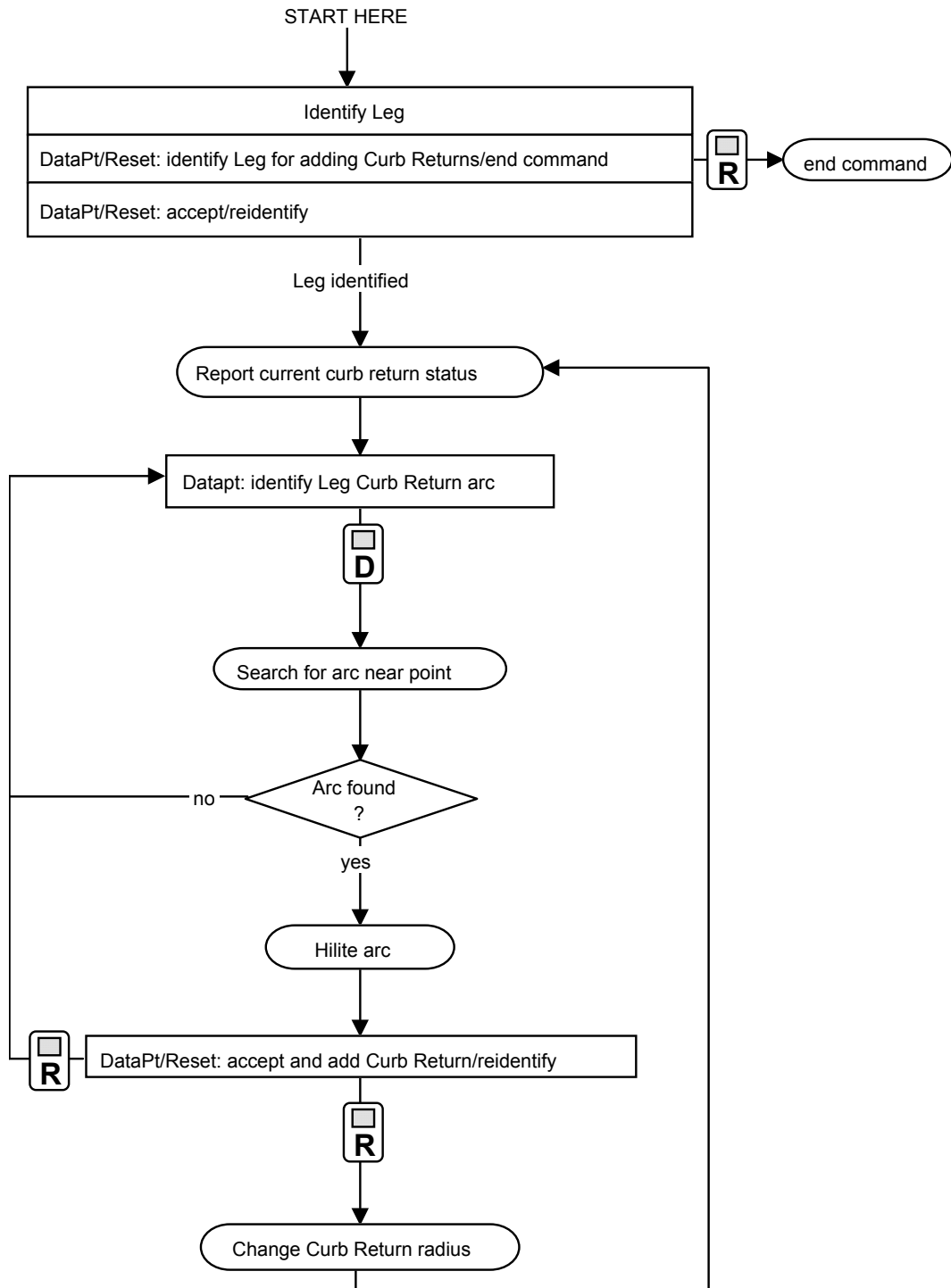
Primitive Command: ADD - LEG CURB CR - BY KEY-IN



Add or revise the Curb Return. A keyed in radius is used for the Curb Return radius. This is the Curb Return between the Inbound Lanes of the identified Leg and the Outbound Lanes of the adjacent Leg.

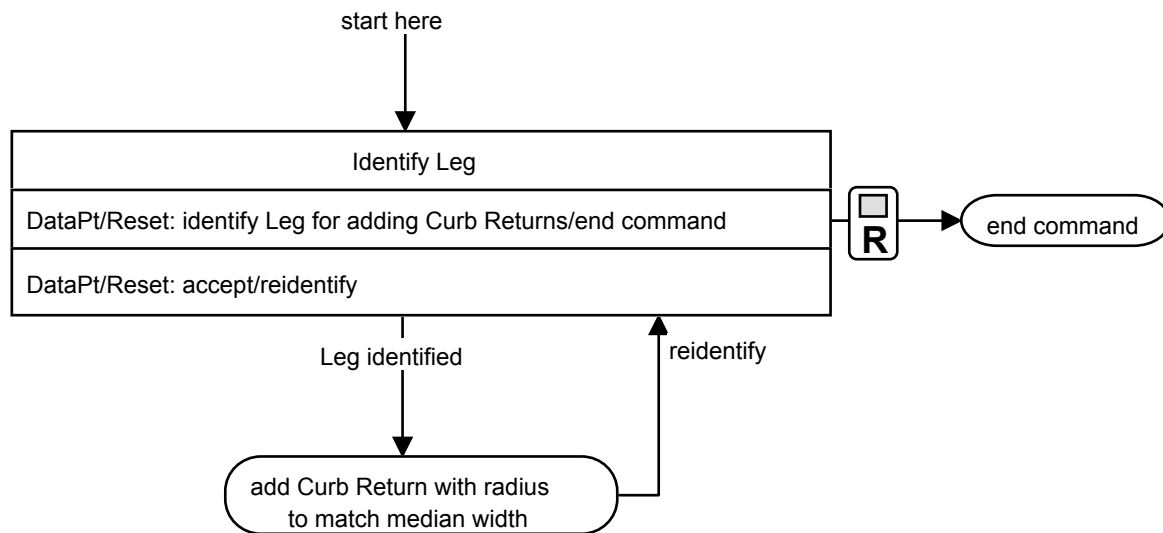
This is the same as **Primitive Command: MODIFY - LEG CURB CR.**

Primitive Command: ADD - LEG CURB CR - SCRATCH LVL



Add or revise the Curb Return. The radius of an identified arc will be used for the Curb Return radius. The arc may not be an IGIDS arc, but must be on the scratch level or in an attached reference file. This is the Curb Return between the Inbound Lanes of the identified Leg and the Outbound Lanes of the adjacent Leg.

Primitive Command: ADD - LEG MEDIAN CR - BY KEY-IN



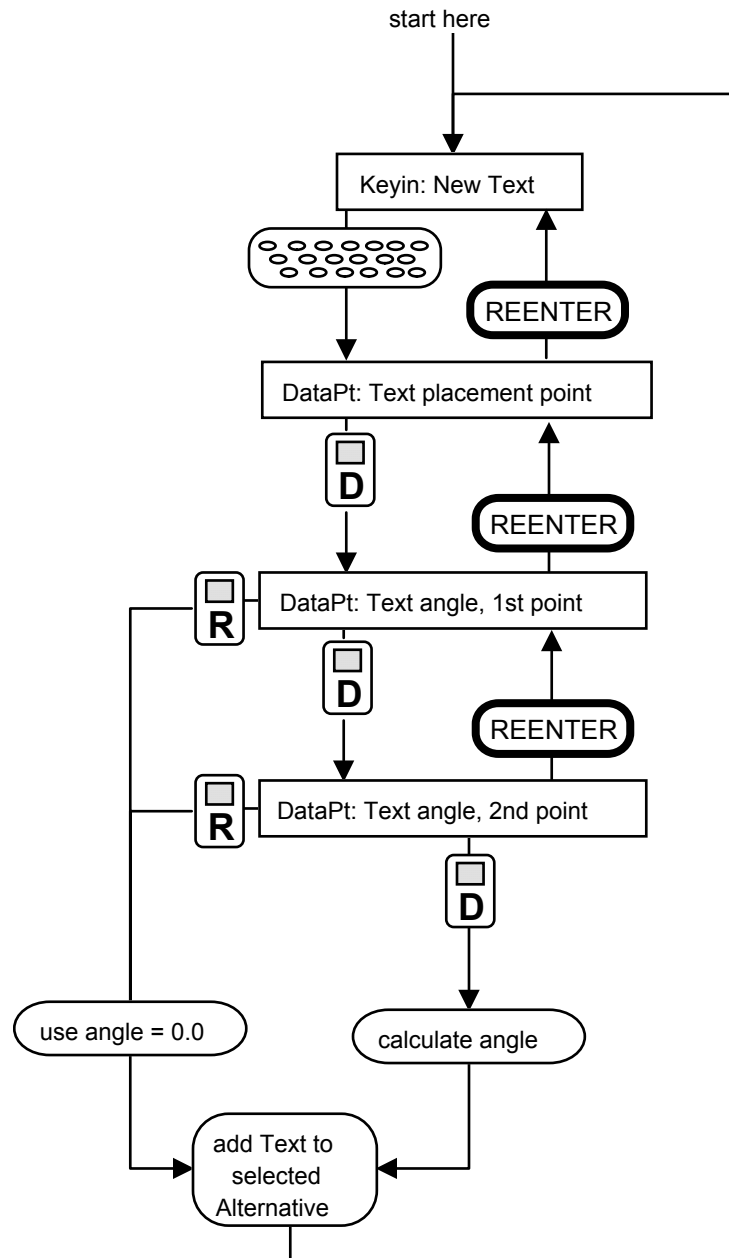
Add the median Curb Return. The Curb Return radius is automatically set to span the median. This is the Curb Return that closes the median between the Inbound Lanes and Outbound Lanes of the identified Leg.

This is the same as **Primitive Command: MODIFY - LEG MEDIAN CR.**

Primitive Command: ADD - LEG MEDIAN CR - SCRATCH LVL

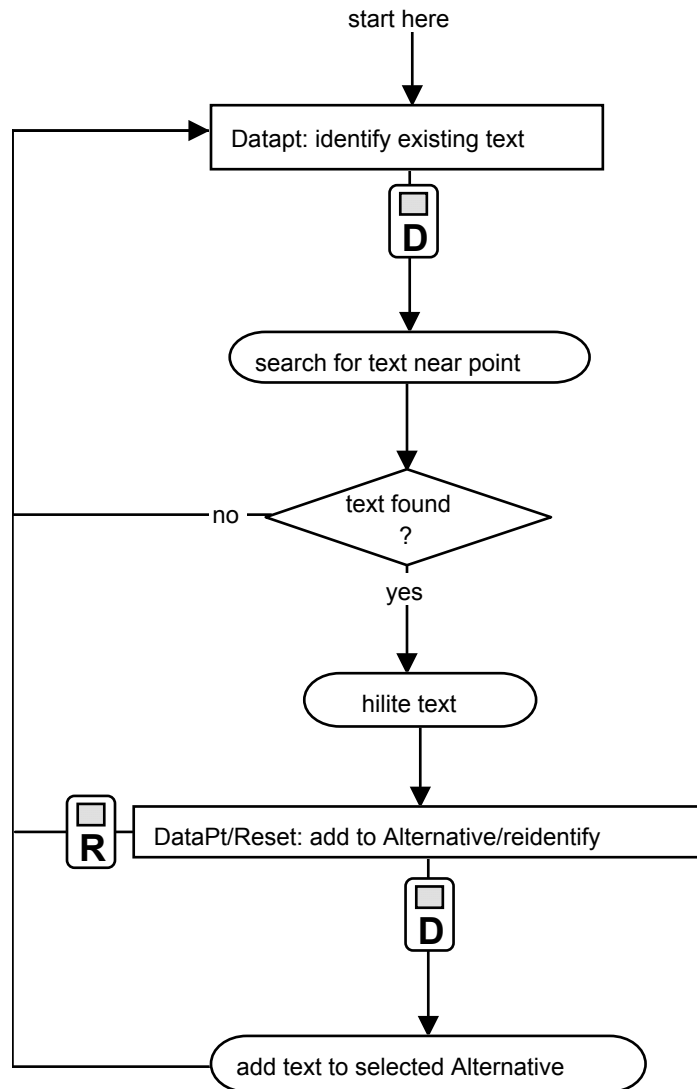
Add the median Curb Return. Not programmed yet.

Primitive Command: ADD - TEXT - TO ALT - BY KEY-IN



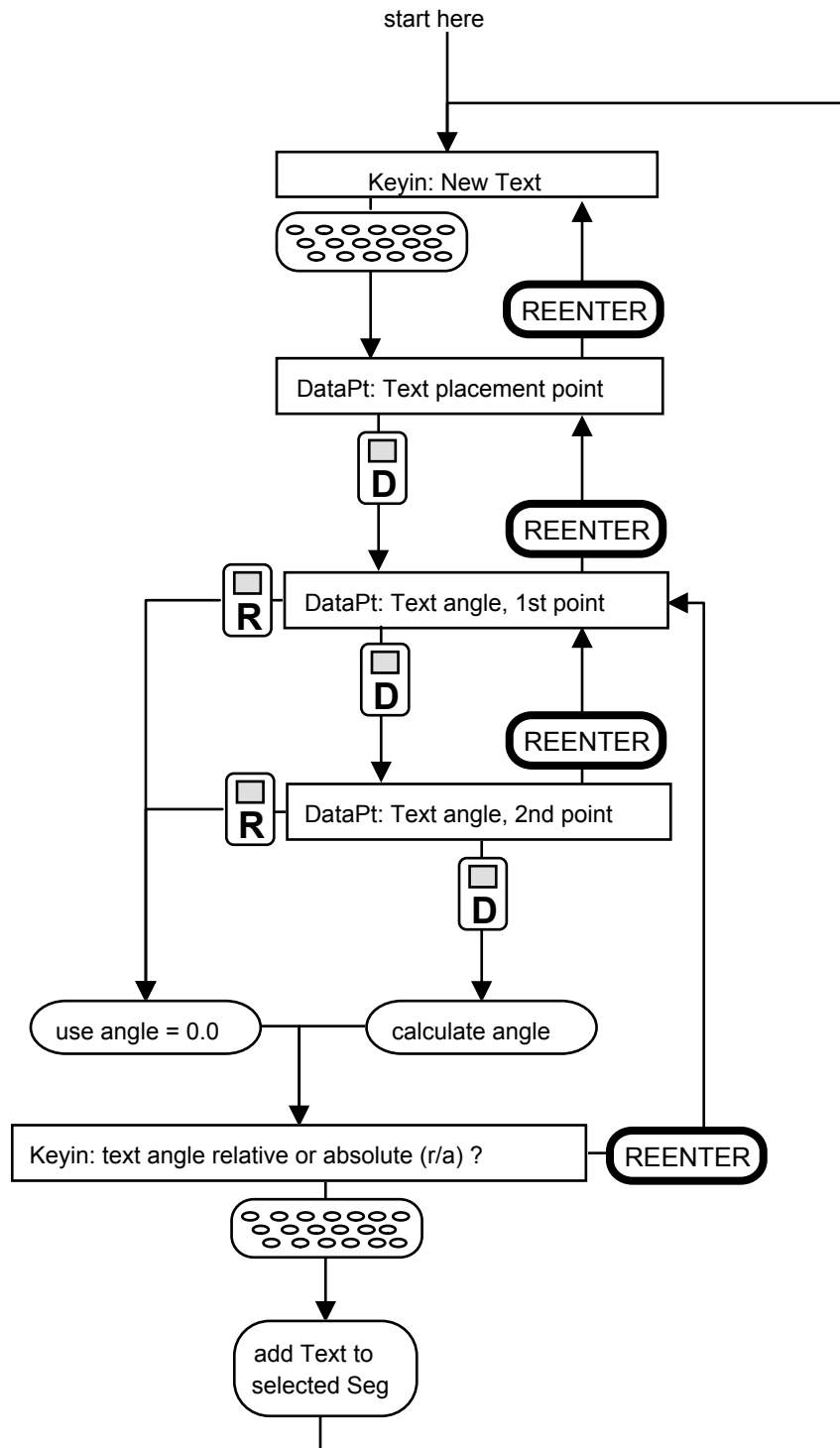
Attach user specified Text to the selected Alternative. The location and absolute angle must be specified by the user. The Graphics Engine's current text size will be used.

Primitive Command: ADD - TEXT - TO ALT - SCRATCH LVL



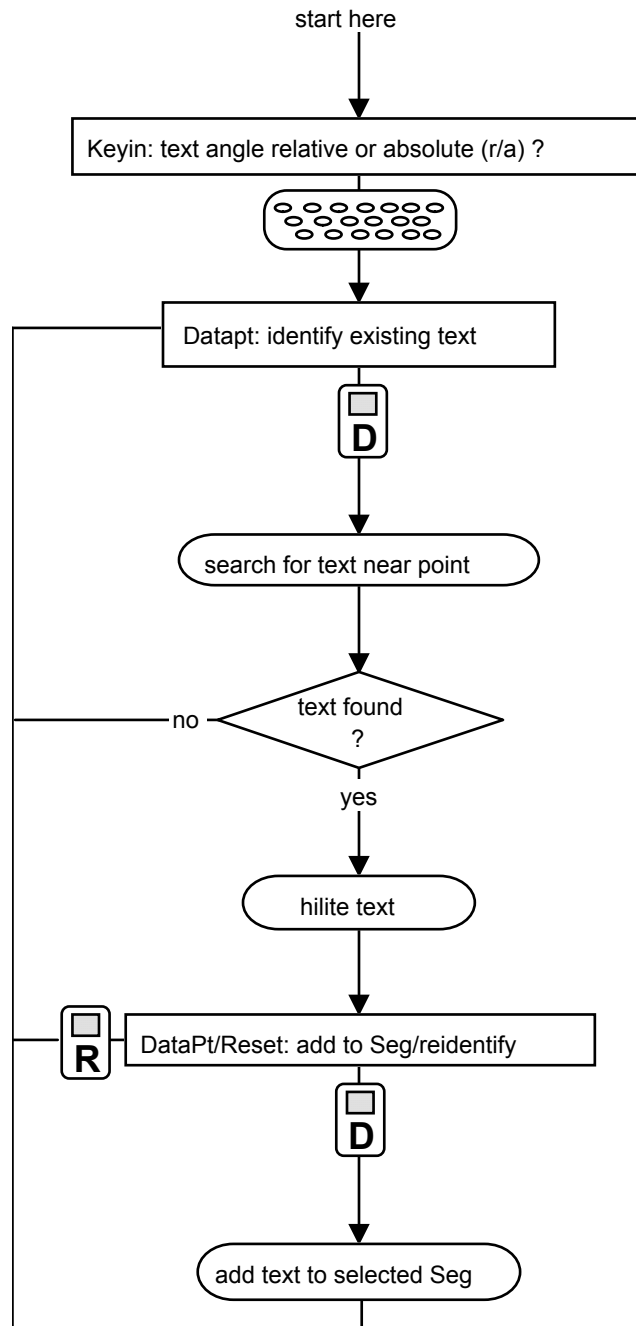
Attach a copy of existing text to the selected Alternative. The existing text cannot be IGIDS Text, but must be on the scratch level or in an attached reference file. The characteristics of the existing text will be used.

Primitive Command: ADD - TEXT - TO SEGMENT - BY KEY-IN



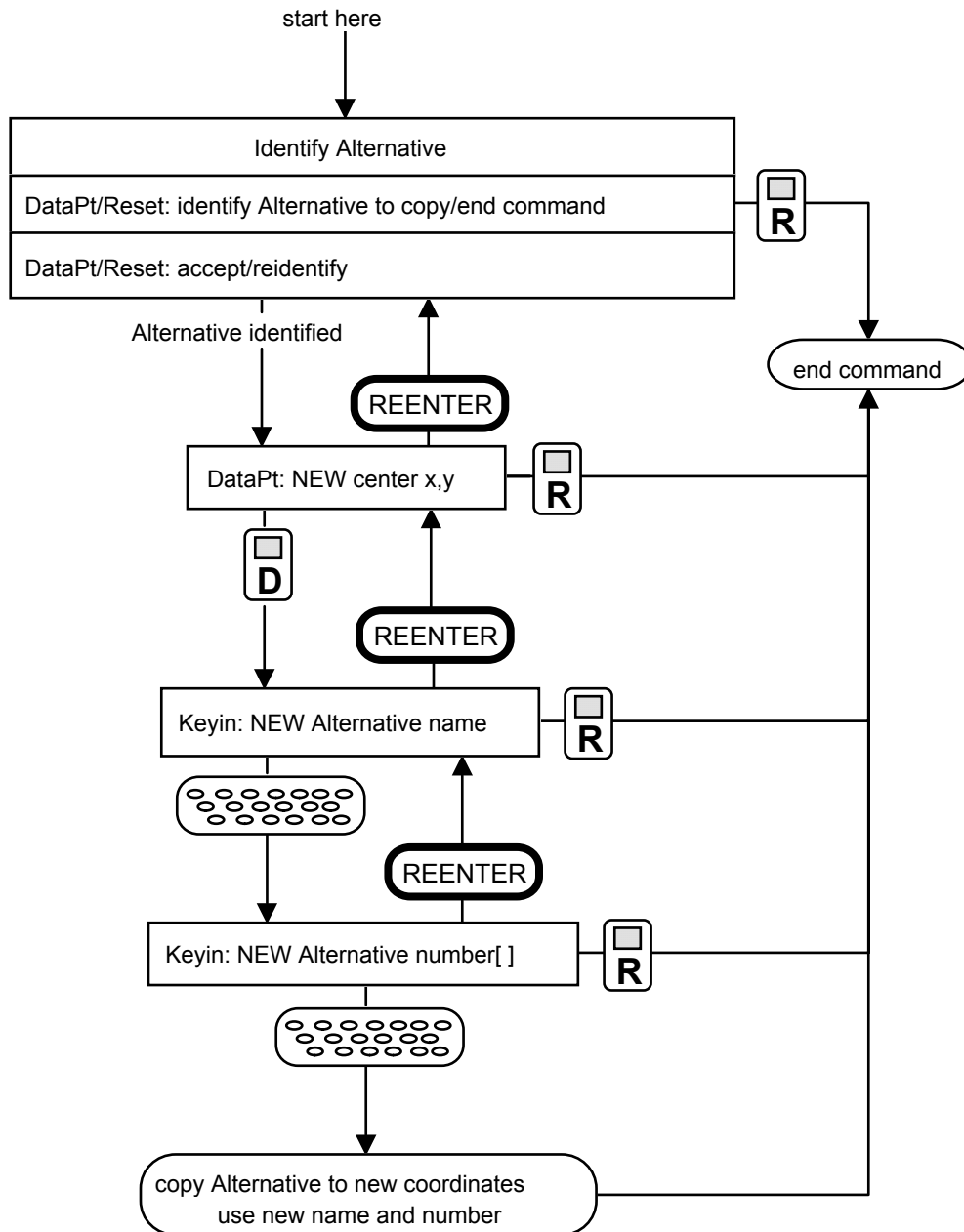
Attach user specified Text to the selected Seg. The location and rotation angle must be specified. The angle may be either absolute or relative. The Graphics Engine's current text size will be used.

Primitive Command: ADD - TEXT - TO SEGMENT - SCRATCH LVL



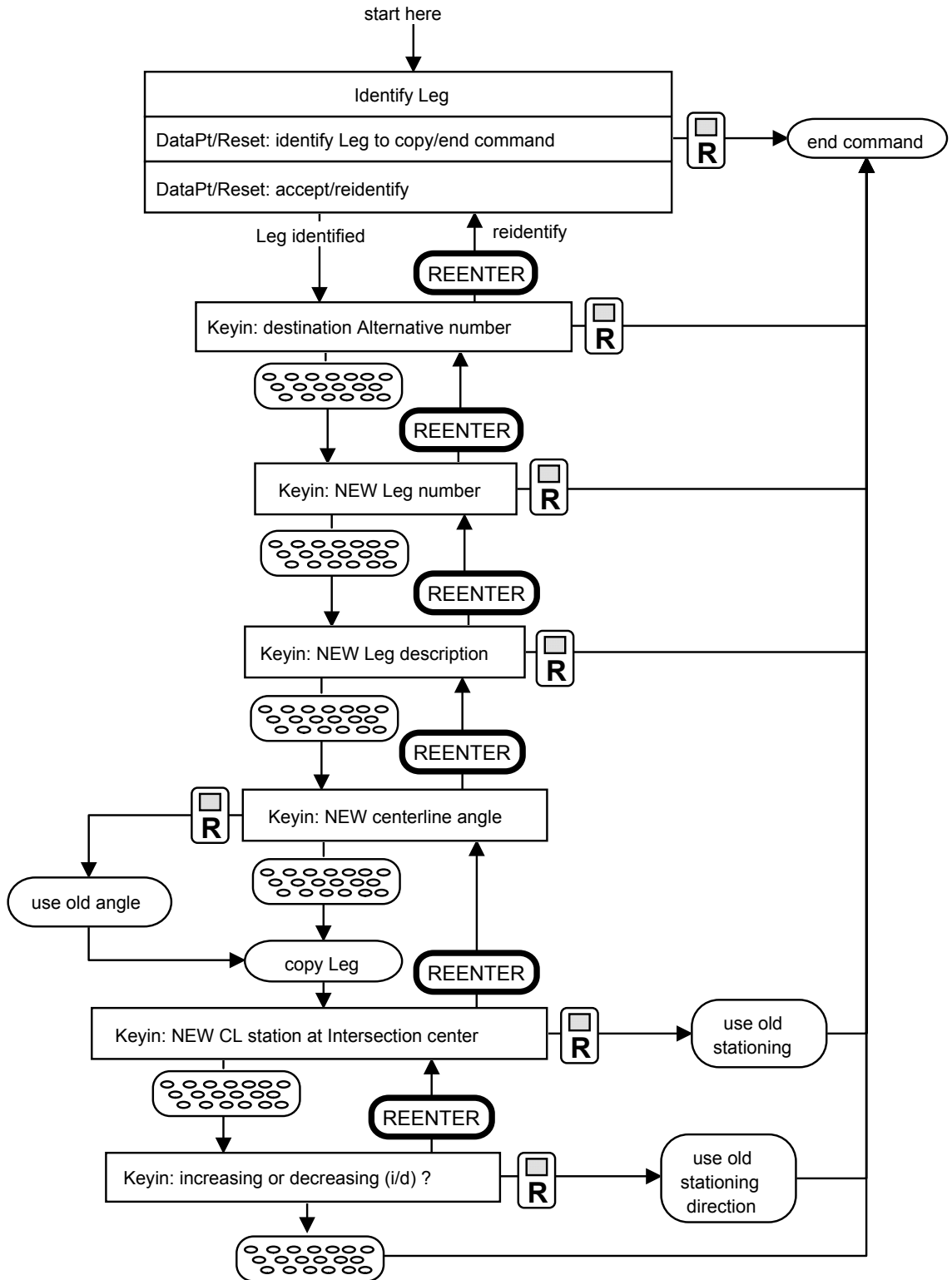
Attach a copy of existing text to the selected Seg. The existing text cannot be IGIDS Text, but must be on the scratch level or in an attached reference file. The location and rotation angle must be specified. The angle may be either absolute or relative.

Primitive Command: COPY- ALTERNATIVE



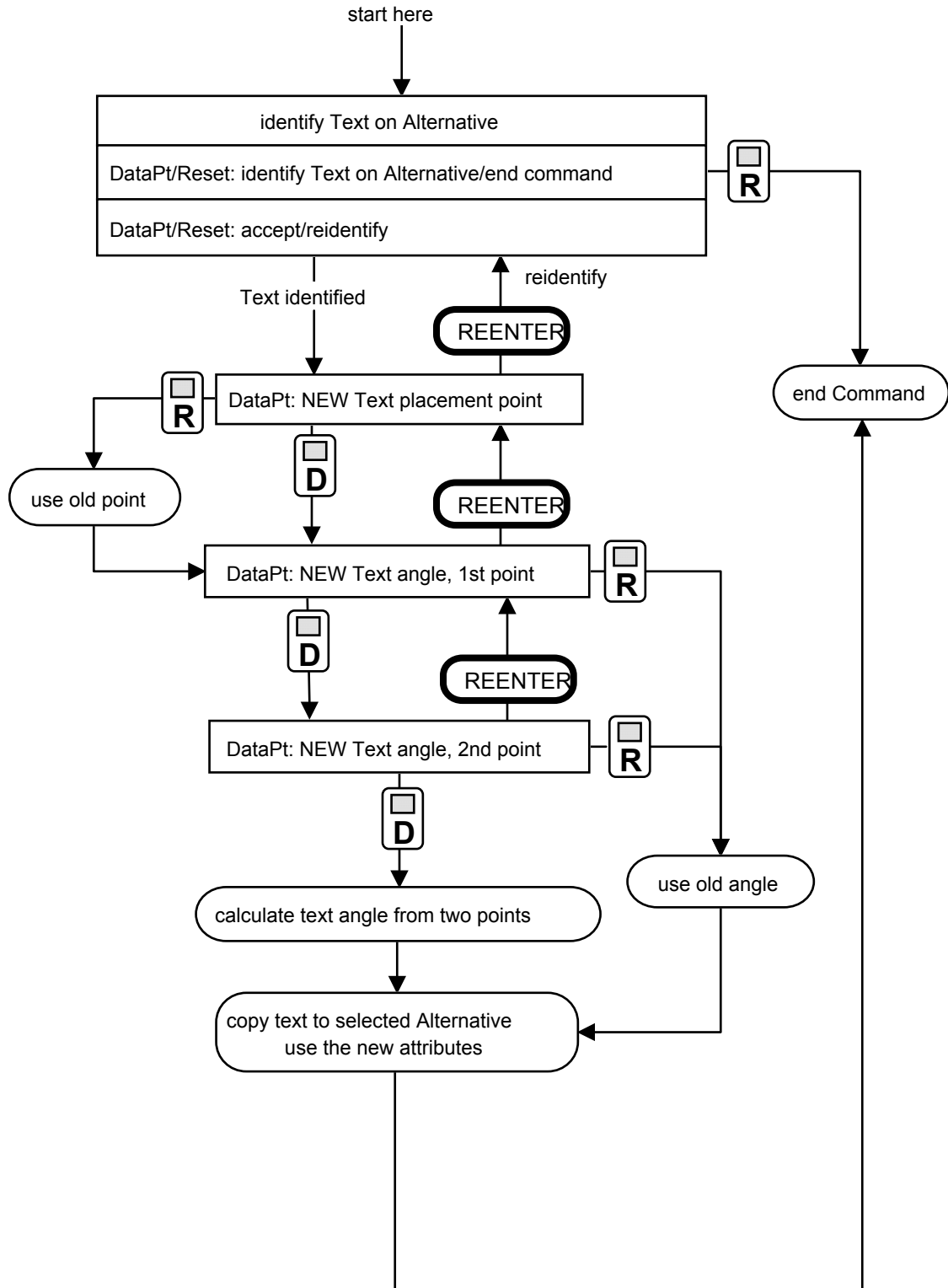
Make a copy of an existing Alternative. New center point coordinates, name, and ID number may be assigned to the new Alternative.

Primitive Command: COPY- LEG



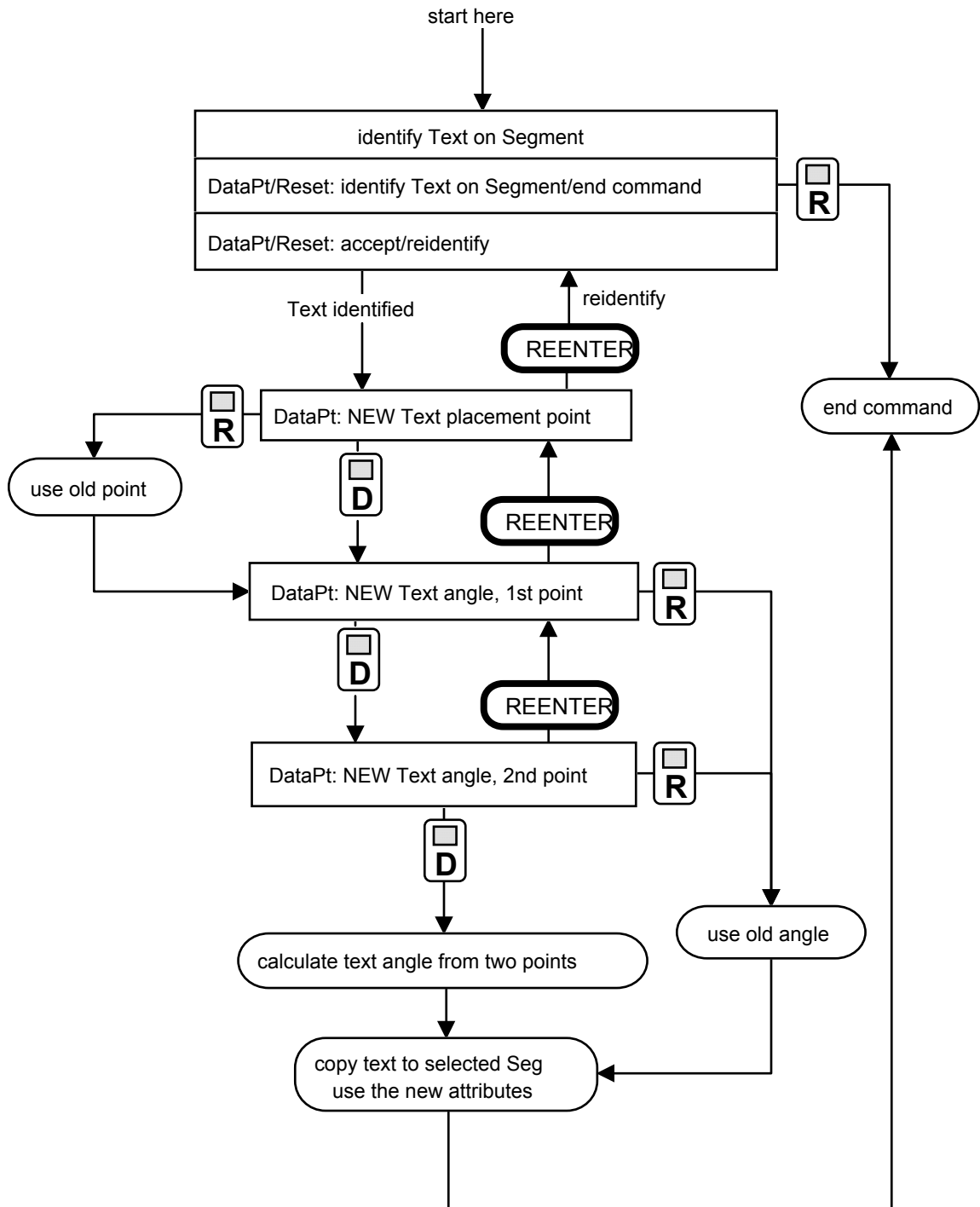
Make a copy of an existing Leg. The Leg may be copied to any Alternative. A new ID number, description, centerline angle, station number at center of intersection and direction of stationing may be assigned to the new Leg.

Primitive Command: COPY - TEXT ON ALT



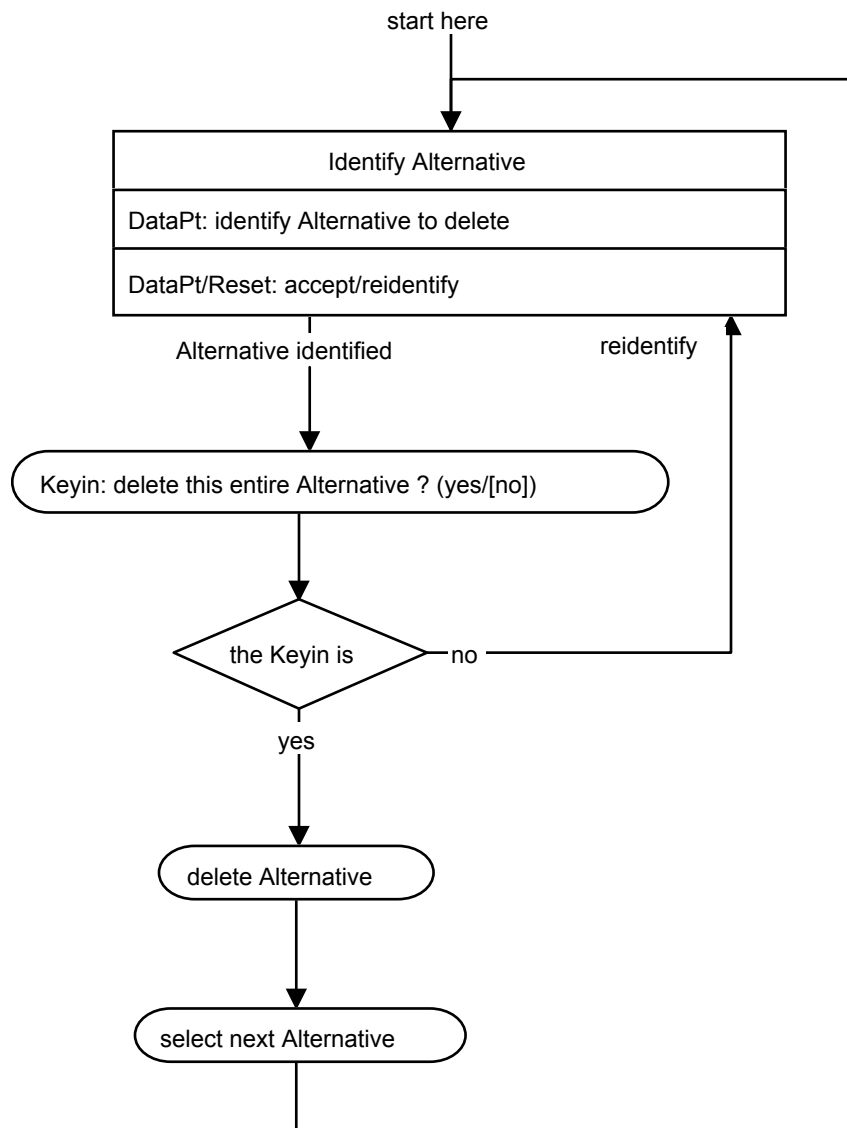
Make a copy of an existing Text on an Alternative and attach it to the selected Alternative. The new Text may have a new location and angle.

Primitive Command: COPY - TEXT ON SEG



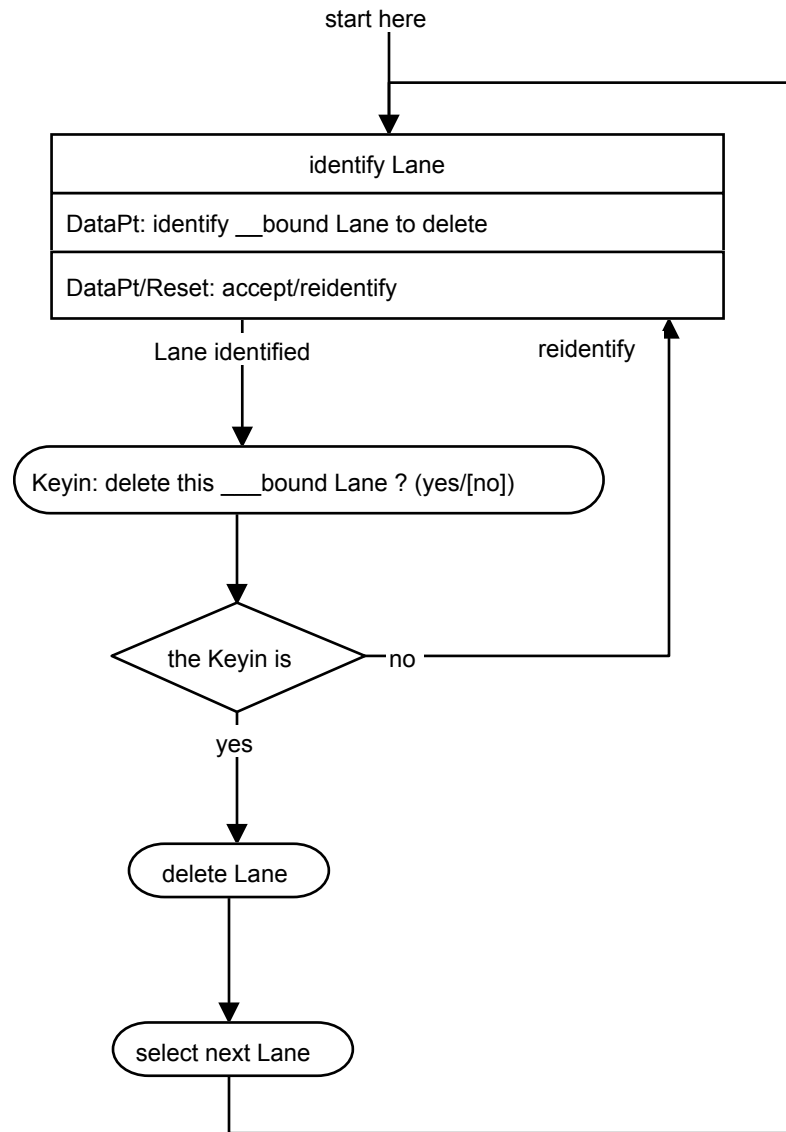
Make a copy of an existing Text on a Seg and attach it to the selected Seg. The new Text may have a new location and angle.

Primitive Command: DELETE - ALTERNATIVE



Delete an existing Alternative. Before each deletion, the user must confirm that the Alternative is to be deleted.

Primitive Command: DELETE - LANE



Use this diagram for the 2 **DELETE - LANE** commands that follow.

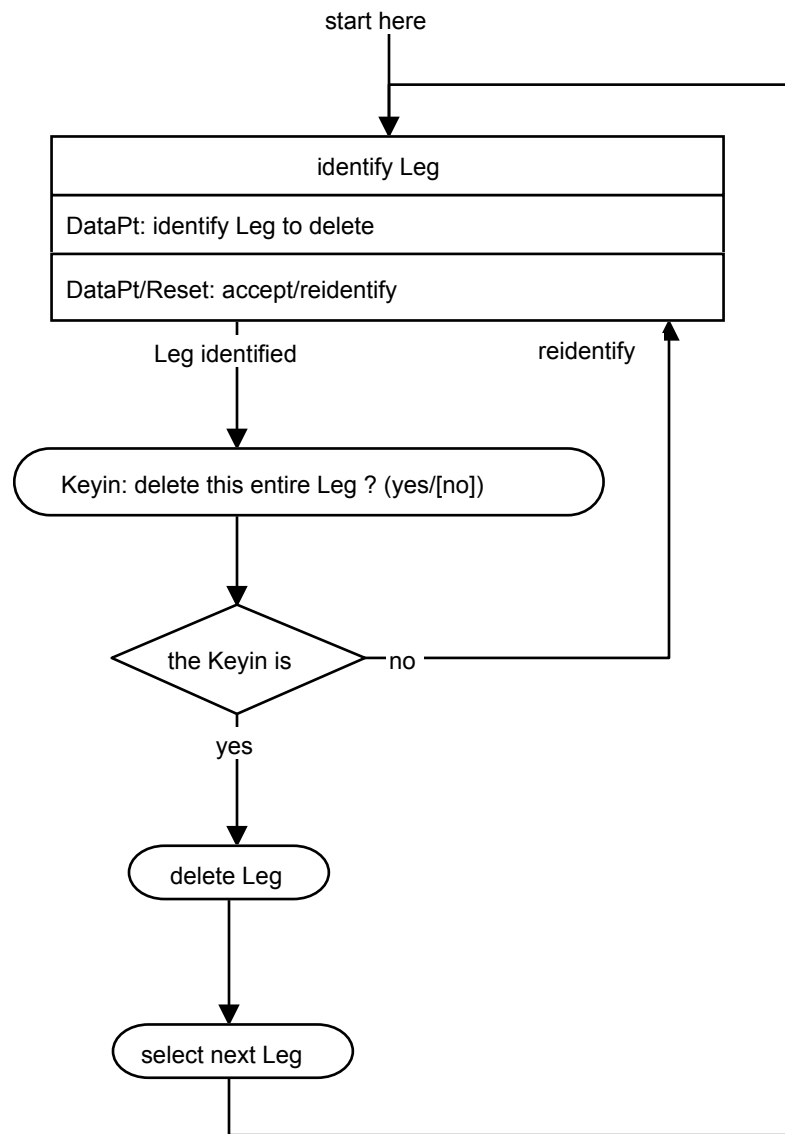
Primitive Command: DELETE - LANE INBND

Delete an existing Inbound Lane. Before each deletion, the user must confirm that the Lane is to be deleted.

Primitive Command: DELETE - LANE OUTBND

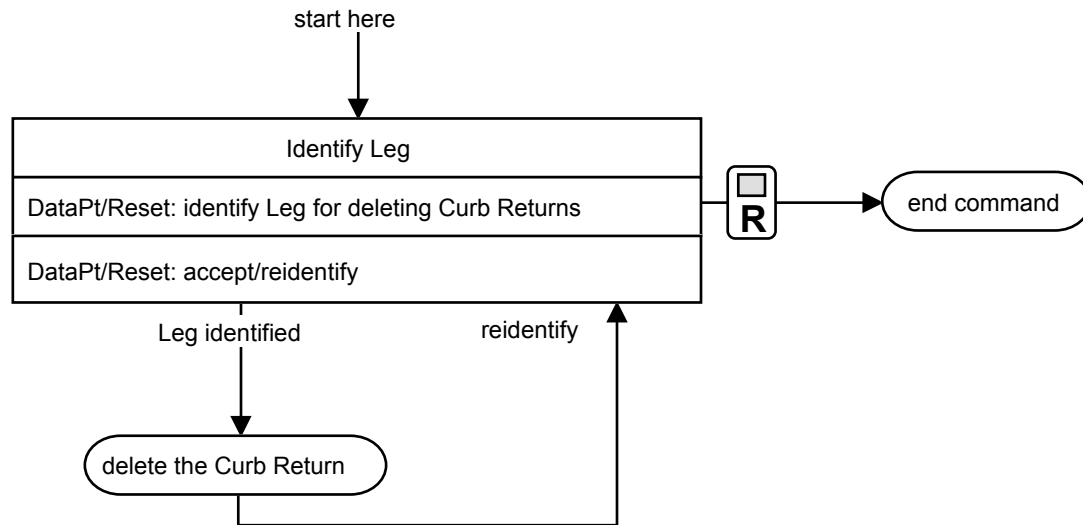
Delete an existing Outbound Lane. Before each deletion, the user must confirm that the Lane is to be deleted.

Primitive Command: DELETE - LEG



Delete an existing Leg. Before each deletion, the user must confirm that the Leg is to be deleted.

Primitive Command: DELETE - LEG - Curb Return



Use this diagram for the 2 **DELETE - LEG - Curb Return** commands that follow.

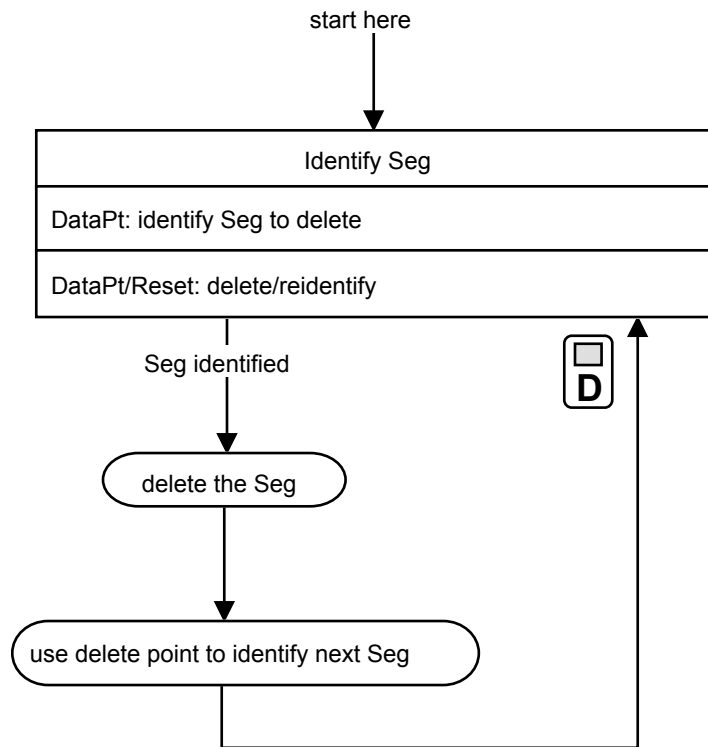
Primitive Command: DELETE - LEG - CURB CR

Delete an existing curb Lane Curb Return. This is the Curb Return between the Inbound Lanes of the identified Leg and the Outbound Lanes of the adjacent Leg.

Primitive Command: DELETE - LEG - MEDIAN CR

Delete an existing median Lane Curb Return. This is the Curb Return that closes the median between the Inbound Lanes and Outbound Lanes of the identified Leg.

Primitive Command: DELETE - SEG



Use this diagram for the 4 **DELETE - SEG** commands that follow. Identify Object will only find Objects of the specific type that the command is designed to delete. For example, it is impossible to identify an Inner Edge Seg when trying to delete a Stop line Seg.

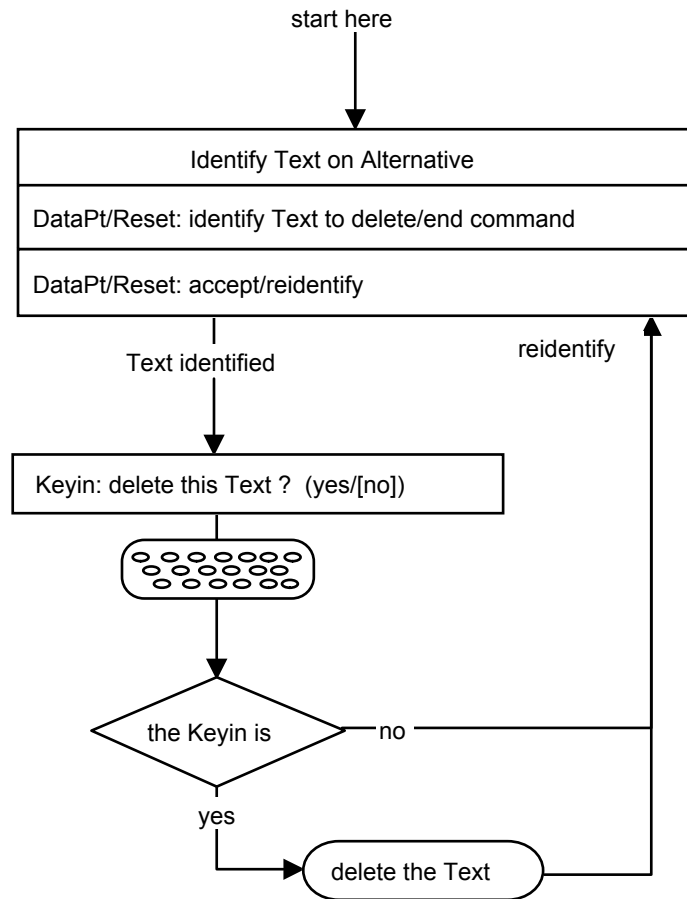
Primitive Command: DELETE - SEG CNTRLIN
Not programmed yet.

Primitive Command: DELETE - SEG INN EDGE
Delete an existing Inner Edge Seg.

Primitive Command: DELETE - SEG OUT EDGE
Delete an existing Outer Edge Seg.

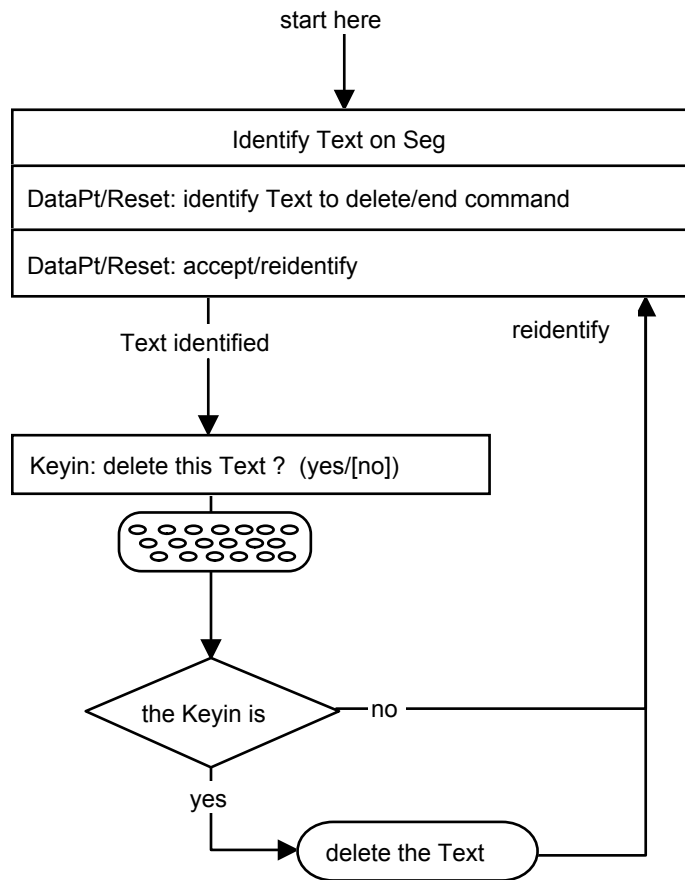
Primitive Command: DELETE - SEG STOPLINE
Delete an existing Stop line Seg.

Primitive Command: DELETE - TEXT ON ALT



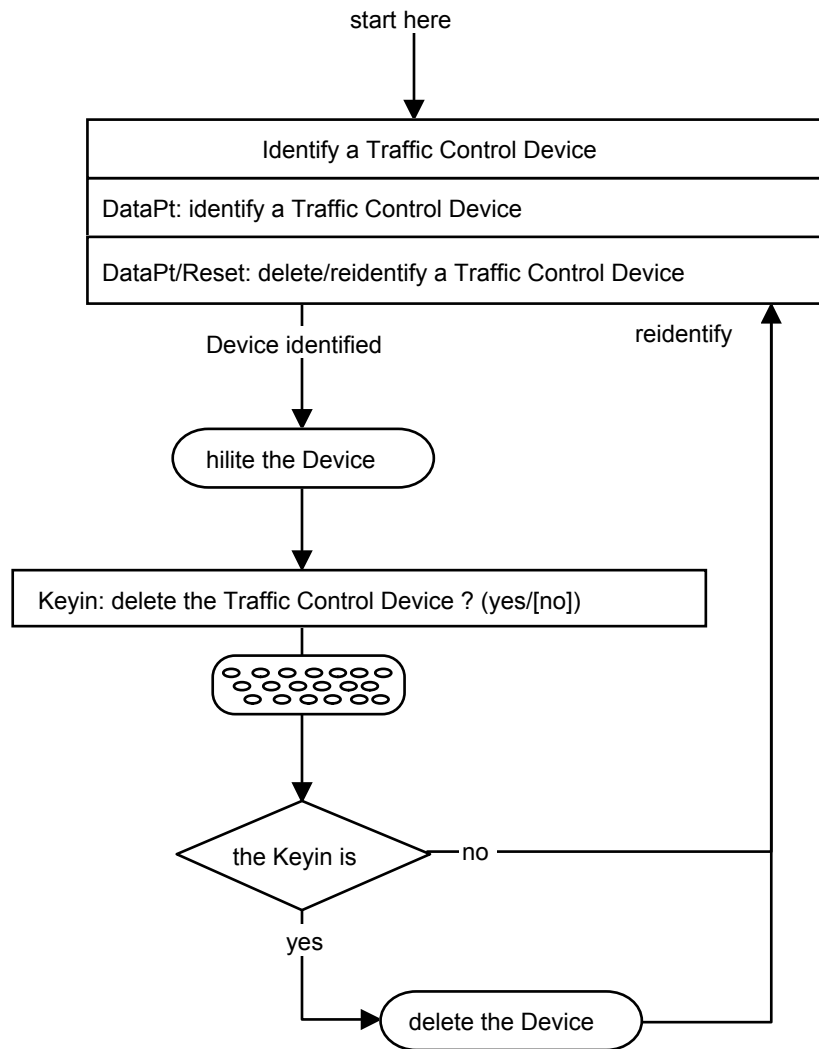
Delete an existing Text on an Alternative. Before each deletion, the user must confirm that the Text is to be deleted.

Primitive Command: DELETE - TEXT ON SEG



Delete an existing Text on a Seg. Before each deletion, the user must confirm that the Text is to be deleted.

Primitive Command: DELETE - TRAF CONTRL



Delete an existing traffic control device. Before each deletion, the user must confirm that the device is to be deleted.

Primitive Command: END IGIDS

Stop IGIDS. IGIDS graphics will remain in the Graphics Engine's database, but the IGIDS data cannot be recreated from this. If IGIDS data has not been saved in some form, it will be lost.

Transient Command: HILITE - CURRENT ALT

Hilite all of the Legs and Text on an Alternative of the selected Alternative.

This is the same as **Transient Command: SELECT - ALTERNATE - CURRENT**

Transient Command: HILITE - CURRENT LANE - ALL

Hilite the Inner Edge Segs, Outer Edge Segs and Stop line Segs of the selected Lane.

This is the same as **Transient Command: SELECT - LANE - CURRENT.**

Transient Command: HILITE - CURRENT LANE - INNER EDGE

Hilite the Inner Edge Segs of the selected Lane.

Transient Command: HILITE - CURRENT LANE - OUTER EDGE

Hilite the Outer Edge Segs of the selected Lane.

Transient Command: HILITE - CURRENT LANE - STOP LINE

Hilite the Stop line Segs of the selected Lane.

Transient Command: HILITE - CURRENT LEG - ALL

Hilite the Inbound Lanes, Outbound Lanes, Centerline Segs, Median Lane Curb Return Segs, and Curb Lane Curb Return Segs of the selected Leg.

This is the same as **Transient Command: SELECT - LEG- CURRENT.**

Transient Command: HILITE - CURRENT LEG - CENTERLINE

Hilite the Centerline Segs of the selected Leg.

Transient Command: HILITE - CURRENT LEG - CURB CR

Hilite the Curb Lane Curb Return Segs of the selected Leg.

Transient Command: HILITE - CURRENT LEG - CURB RETURNS

Hilite the Median Lane Curb Return Segs and Curb Lane Curb Return Segs of the selected Leg.

Transient Command: HILITE - CURRENT LEG - INBND LANES

Hilite the Inbound Lanes of the selected Leg.

Transient Command: HILITE - CURRENT LEG - MEDIAN CR

Hilite the Median Lane Curb Return Segs of the selected Leg.

Transient Command: HILITE - CURRENT LEG - OUTBND LANES

Hilite the Outbound Lanes of the selected Leg.

Transient Command: HILITE - CURRENT SEG

Hilite the selected Seg.

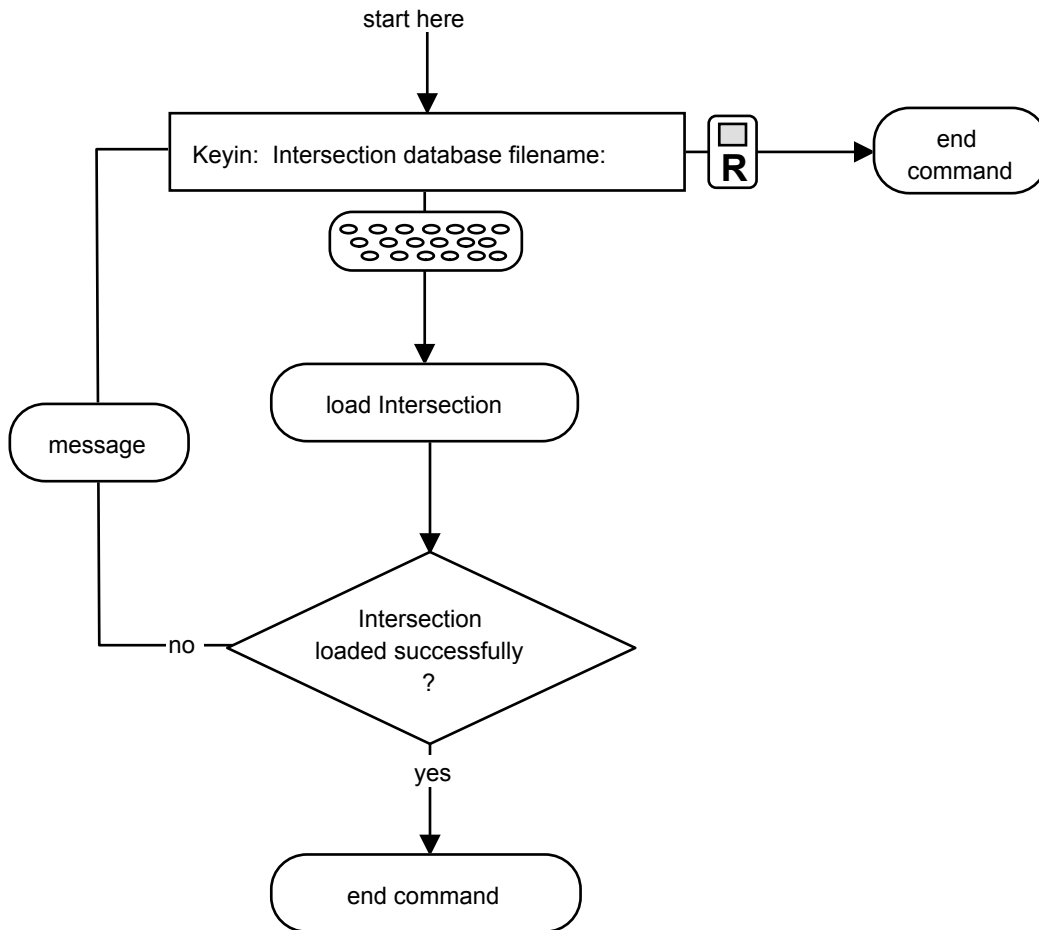
This is the same as **Transient Command: SELECT - SEGMENT - CURRENT.**

Transient Command: HILITE - CURRENT TEXT

Hilite the selected Text.

This is the same as **Transient Command: SELECT - TEXT - CURRENT.**

Primitive Command: LOAD FROM - DATABASE



Load Intersection data from a file that was written by **Primitive Command: SAVE TO - Data Base**. All data from the current IGIDS session will be lost.

Primitive Command: LOAD FROM - GEOPAK - Centerline

Keyin: Alternative name

keyin: characters

datapt: not allowed

reset: not allowed

reenter: prompt again

Keyin: Alternative number[lowest unused alternative number]

keyin: integer; if already in use then error message then prompt again

datapt: not allowed

reset: not allowed

reenter: backup to Keyin: Alternative name

identify a centerline segment

accept centerline segment

arc or line

if element not from a reference file then error message and prompt again

get class, color, level, style, and weight from selected element and set as default

set search criteria to include default class, color, level, style, and weight

open dialog box for class, color, level, style, and weight

upon OK, read all elements from the reference file matching class, color, level, style, and weight

criteria and add as centerline segments; if all matching elements are not processed then

error message and end command; if no segments added then error message and prompt

again

upon Cancel, prompt again

add alternative, add legs, and add leg centerline segments

Primitive Command: LOAD FROM - GEOPAK - Lane Edges

if no alternative selected then error message and end command
if selected alternative not completed then error message and end command
if selected alternative has no legs then error message and end command
if any leg of selected alternative has inbound or outbound lanes then error message and end command

Keyin: Dist. from CL to closest lane edge [0.0]

keyin: double; if < 0.0 or $>$ maximum lane width * maximum number of lanes then error message and prompt again

datapt: not allowed

reset: not allowed

reenter: prompt again

Keyin: Max curb return radius [program value]

keyin: double; if ≤ 0.0 then error message and prompt again

datapt: not allowed

reset: backup to Keyin: Dist. from CL to closest lane edge

reenter: prompt again

identify a lane edges segment

accept lane edges segment

arc or line

if element not from a reference file then error message and prompt again

get class, color, level, style, and weight from selected element and set as default

set search criteria to include default class, color, level, style, and weight

open dialog box for class, color, level, style, and weight

upon OK, read all elements from the reference file matching class, color, level, style, and weight criteria and add as lane edge segments for leg closest to selected element; if all matching elements are not processed then error message and end command; if no segments added then error message and prompt again

upon Cancel, prompt again

add lanes, add lane inner edge segments, add lane outer edge segments, add stop line segments, and add curb returns

Primitive Command: LOAD FROM - GEOPAK - Pavement Edges

if no alternative selected then error message and end command

if selected alternative has no legs then error message and end command

if all legs of selected alternative have inbound and outbound lanes then error message and end command

Keyin: Dist. from CL to closest lane edge [0.0]

keyin: double; if < 0.0 or > maximum lane width * maximum number of lanes then error message and prompt again

datapt: not allowed

reset: not allowed

reenter: prompt again

Keyin: Max curb return radius [program value]

keyin: double; if <= 0.0 then error message and prompt again

datapt: not allowed

reset: backup to Keyin: Dist. from CL to closest lane edge

reenter: prompt again

Keyin: L/W for # Lanes/lane Width

keyin: character

datapt: not allowed

reset: backup to Keyin: Max curb return radius

reenter: prompt again

if L then Keyin: Number of lanes (1-max lanes per leg) [1]

keyin: integer; if < 1 or > max lanes per leg then error message and prompt again

datapt: not allowed

reset: backup to Keyin: L/W for # Lanes/lane Width

reenter: prompt again

if W then Keyin: Lane width (minimum lane width-maximum lane width) [default lane width]

keyin: double; if < minimum lane width or > maximum lane width then error message and prompt again

datapt: not allowed

reset: backup to Keyin: L/W for # Lanes/lane Width

reenter: prompt again

identify a pavement edges segment

accept pavement edges segment

arc or line

if element not from a reference file then error message and prompt again

get class, color, level, style, and weight from selected element and set as default

set search criteria to include default class, color, level, style, and weight

open dialog box for class, color, level, style, and weight

upon OK, read all elements from the reference file matching class, color, level, style, and weight

criteria and add as lane edge segments for all legs; if L then for each leg inbound and outbound lanes set lane width = distance between median lane inner edge and curb lane

outer edge / number of lanes and add additional lanes and lane edges as needed; if W

then for each leg inbound and outbound lanes set number of lanes rounded to the

nearest integer = distance between median lane inner edge and curb lane outer edge /

lane width then set lane width = distance between median lane inner edge and curb lane

outer edge / number of lanes and add additional lanes and lane edges as needed; if all

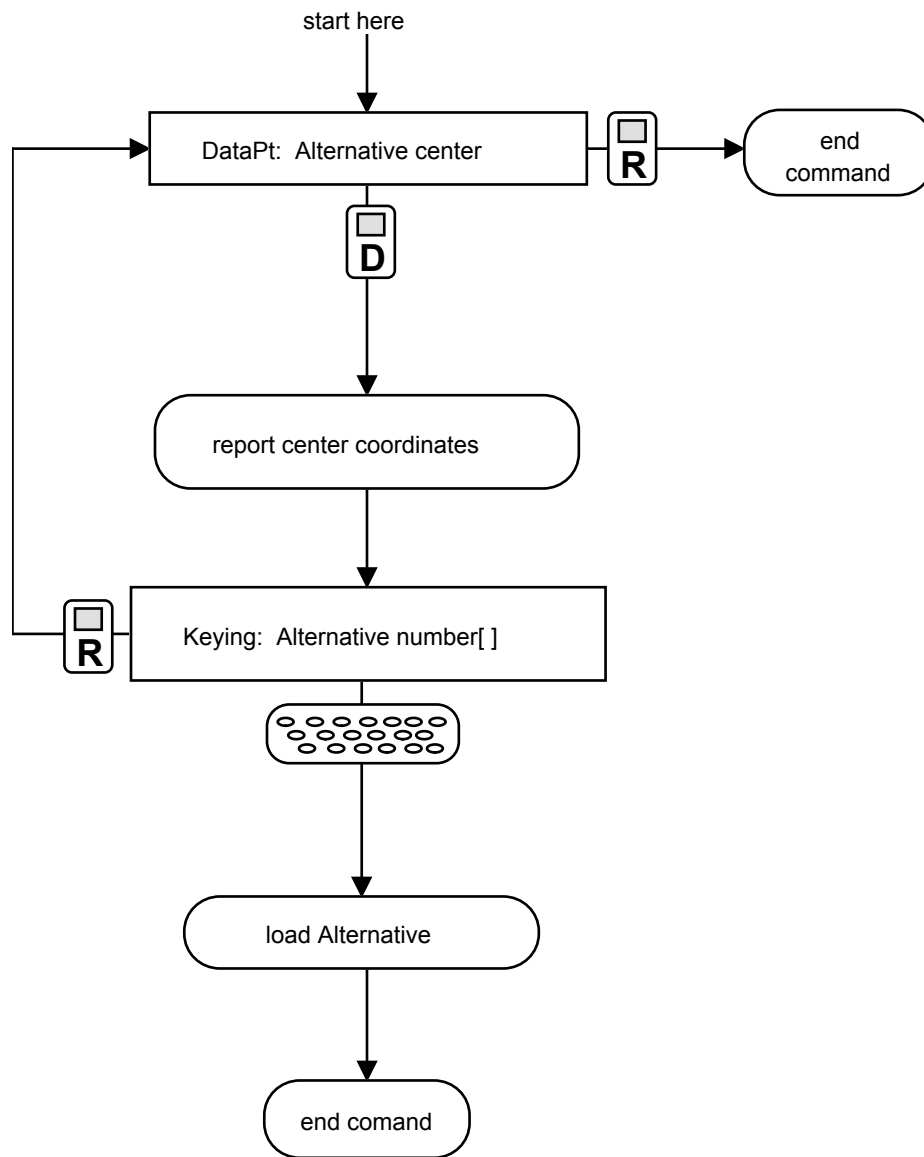
matching elements are not processed then error message and end command; if no

segments added then error message and prompt again

upon Cancel, prompt again

add lanes, add lane inner edge segments, add lane outer edge segments, add stop line segments, and add curb returns

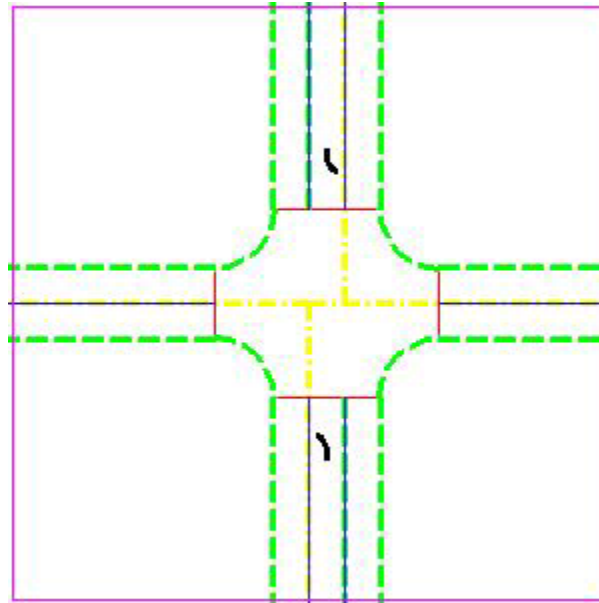
Primitive Command: LOAD FROM - STANDARD



Use this diagram for the 17 **LOAD FROM - STANDARD** commands that follow.

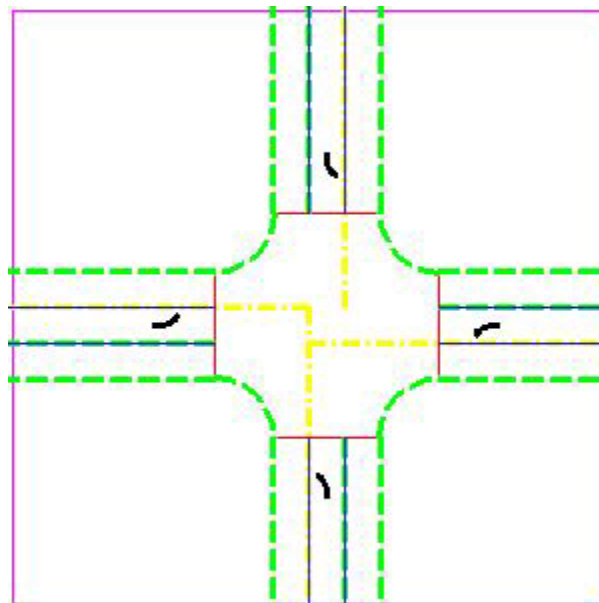
Primitive Command: LOAD FROM - STANDARD - 3X2

Load a standard 4 leg Alternative that has 1 through Inbound Lane, 1 exclusive left turn Inbound Lane, and 1 Outbound Lane on the north-south street and 1 through Inbound Lane and 1 Outbound Lane on the east-west street. Coordinates of the center point and the ID number must be specified.



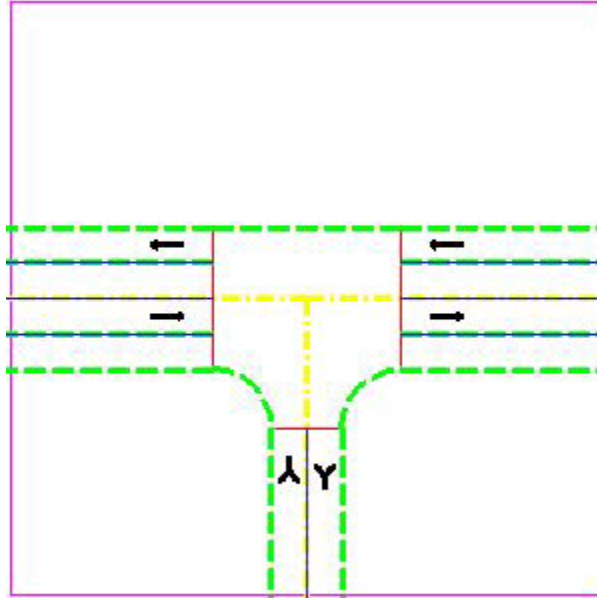
Primitive Command: LOAD FROM - STANDARD - 3X3

Load a standard 4 leg Alternative that has 1 through Inbound Lane, 1 exclusive left turn Inbound Lane, and 1 Outbound Lane in each direction. Coordinates of the center point and the ID number must be specified.



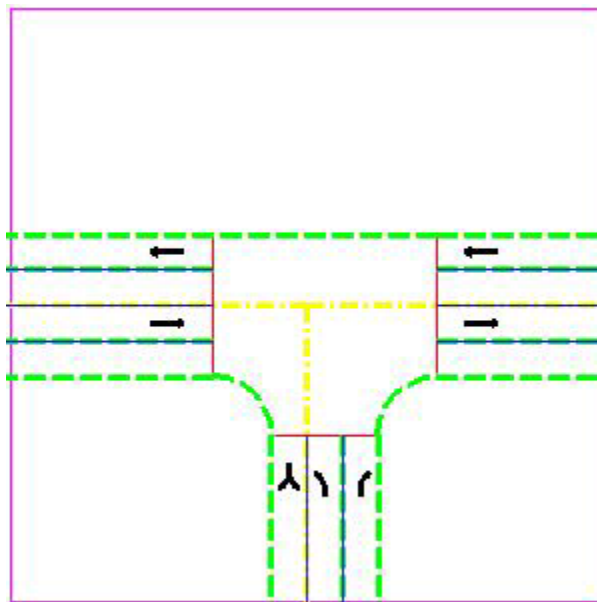
Primitive Command: LOAD FROM - STANDARD - 4T2

Load a standard 3 leg "T" Alternative. The northbound approach T's into the east-west street. The northbound approach has 1 Inbound Lane and 1 Outbound Lane. The east-west street has 2 through Inbound Lanes and 2 Outbound Lanes. Coordinates of the center point and the ID number must be specified.



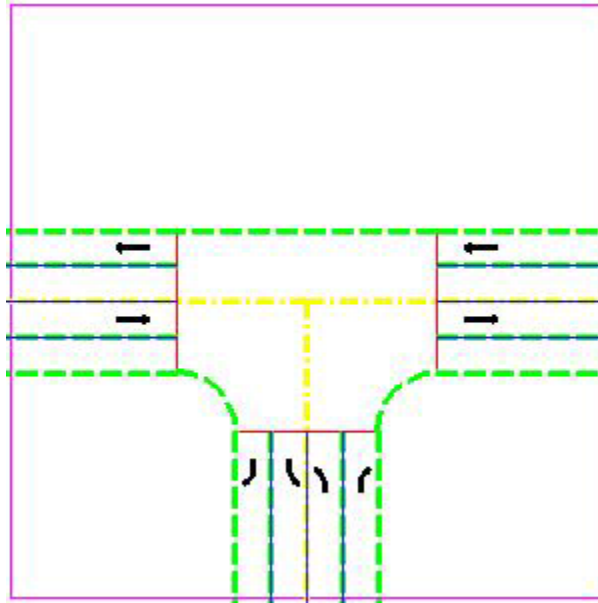
Primitive Command: LOAD FROM - STANDARD - 4T3

Load a standard 3 leg "T" Alternative. The northbound approach T's into the east-west street. The northbound approach has 2 Inbound Lanes and 1 Outbound Lane. The east-west street has 2 Inbound Lanes and 2 Outbound Lanes. Coordinates of the center point and the ID number must be specified.



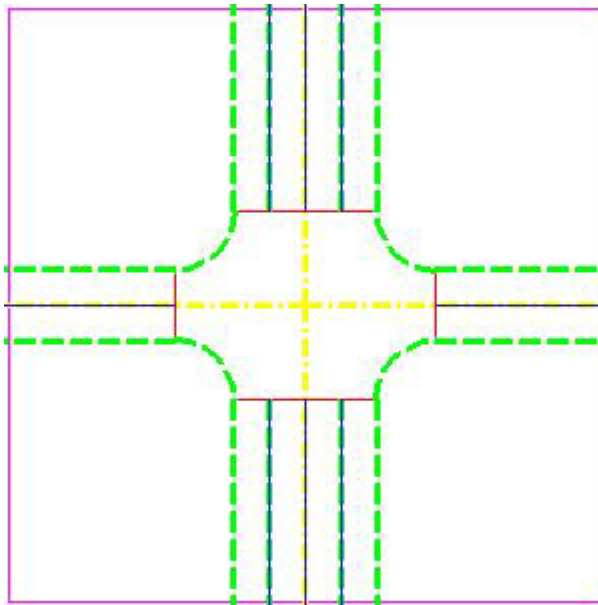
Primitive Command: LOAD FROM - STANDARD - 4T4

Load a standard 3 leg "T" Alternative. The northbound approach T's into the east-west street. Each leg has 2 Inbound Lanes and 2 Outbound Lanes. Coordinates of the center point and the ID number must be specified.



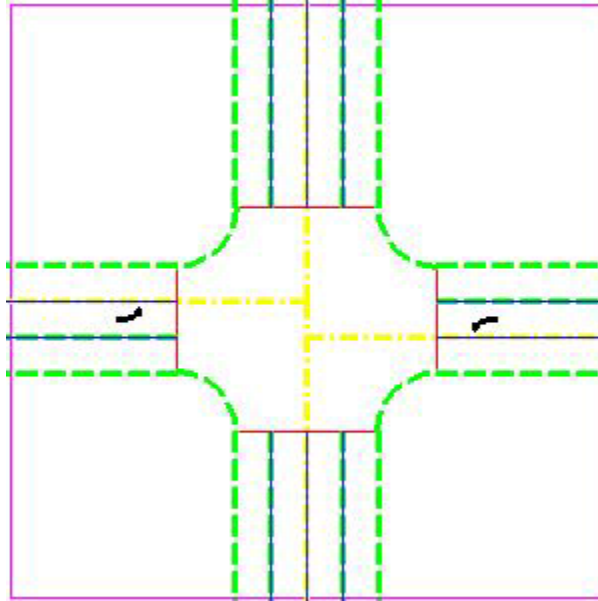
Primitive Command: LOAD FROM - STANDARD - 4X2

Load a standard 4 leg Alternative that has 2 through Inbound Lanes and 2 Outbound Lanes on the north-south street and 1 through Inbound Lane and 1 Outbound Lane on the east-west street. Coordinates of the center point and the ID number must be specified.



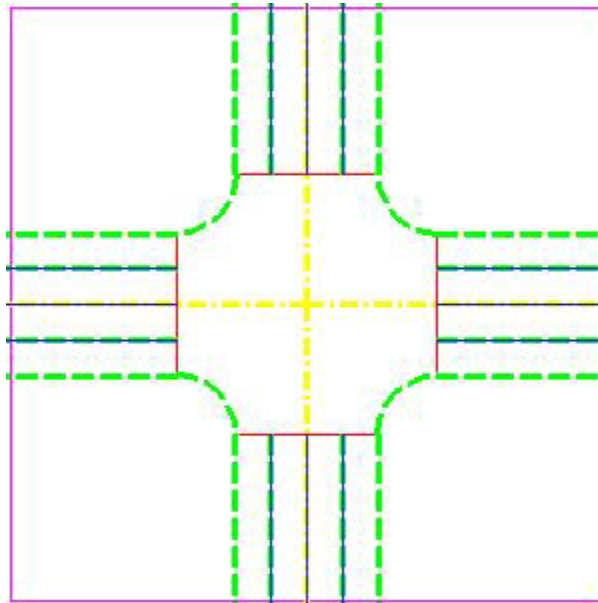
Primitive Command: LOAD FROM - STANDARD - 4X3

Load a standard 4 leg Alternative that has 2 through Inbound Lanes and 2 Outbound Lanes on the north-south street and 1 through Inbound Lane, 1 exclusive left turn Inbound Lane, and 1 Outbound Lane on the east-west street. Coordinates of the center point and the ID number must be specified.



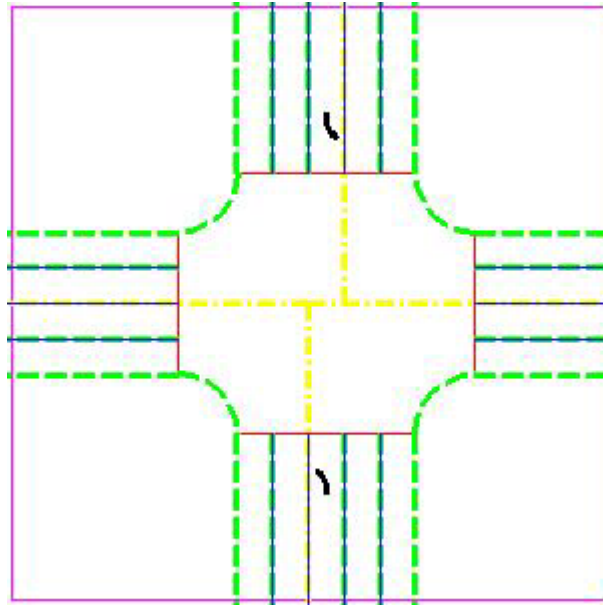
Primitive Command: LOAD FROM - STANDARD - 4X4

Load a standard 4 leg Alternative that has 2 through Inbound Lanes and 2 Outbound Lanes in each direction. Coordinates of the center point and the ID number must be specified.



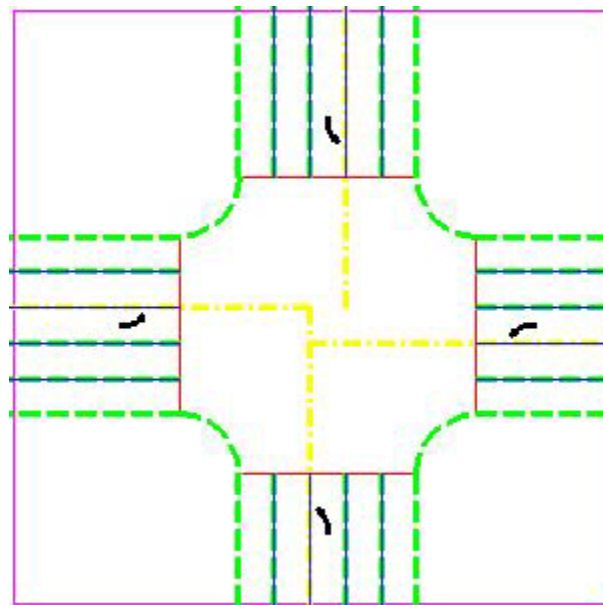
Primitive Command: LOAD FROM - STANDARD - 5X4

Load a standard 4 leg Alternative that has 2 through Inbound Lanes, 1 exclusive left turn Inbound Lane, and 2 Outbound Lanes on the north-south street and 2 through Inbound Lanes and 2 Outbound Lanes on the east-west street. Coordinates of the center point and the ID number must be specified.



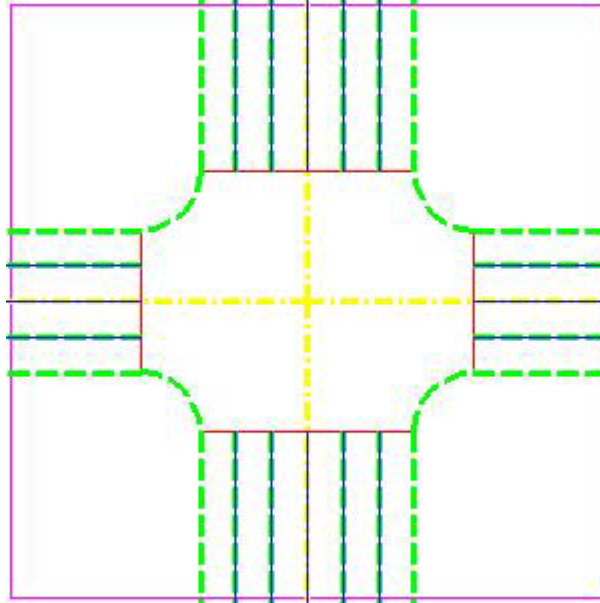
Primitive Command: LOAD FROM - STANDARD - 5X5

Load a standard 4 leg Alternative that has 2 through Inbound Lanes, 1 exclusive left turn Inbound Lane, and 2 Outbound Lanes in each direction. Coordinates of the center point and the ID number must be specified.



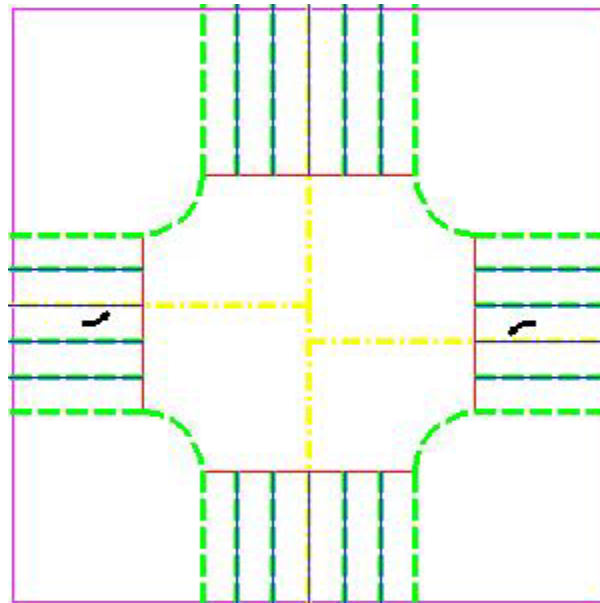
Primitive Command: LOAD FROM - STANDARD - 6X4

Load a standard 4 leg Alternative that has 3 through Inbound Lanes and 3 Outbound Lanes on the north-south street and 2 through Inbound Lanes and 2 Outbound Lanes on the east-west street. Coordinates of the center point and the ID number must be specified.



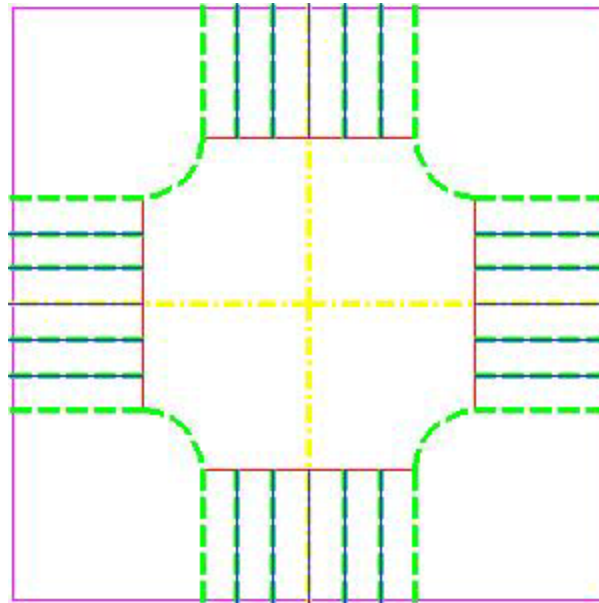
Primitive Command: LOAD FROM - STANDARD - 6X5

Load a standard 4 leg Alternative that has 3 through Inbound Lanes and 3 Outbound Lanes on the north-south street and 2 through Inbound Lanes, 1 exclusive left turn Inbound Lane, and 3 Outbound Lanes on the east-west street. Coordinates of the center point and the ID number must be specified.



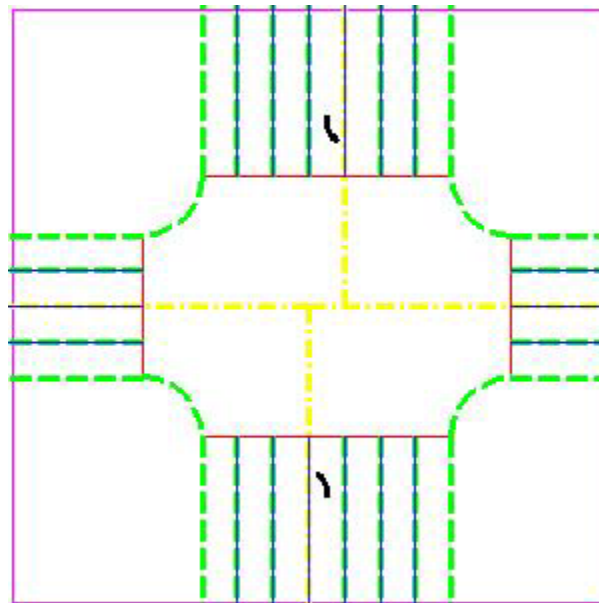
Primitive Command: LOAD FROM - STANDARD - 6X6

Load a standard 4 leg Alternative that has 3 through Inbound Lanes and 3 Outbound Lanes in each direction. Coordinates of the center point and the ID number must be specified.



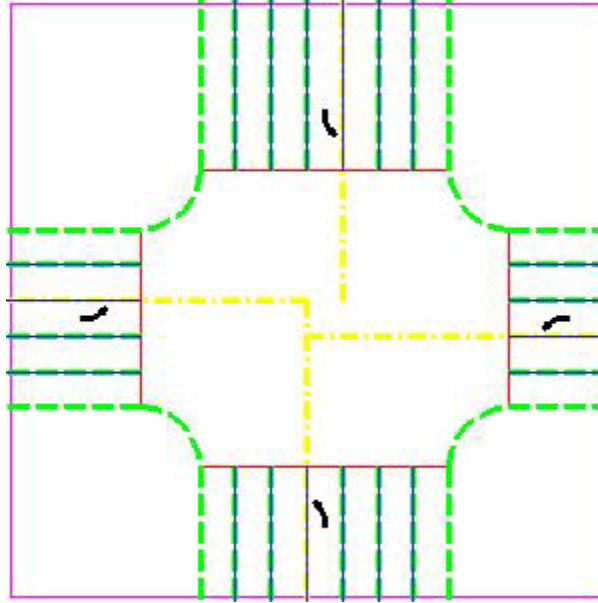
Primitive Command: LOAD FROM - STANDARD - 7X4

Load a standard 4 leg Alternative that has 3 through Inbound Lanes, 1 exclusive left turn Inbound Lane, and 3 Outbound Lanes on the north-south street and 2 through Inbound Lanes and 2 Outbound Lanes on the east-west street. Coordinates of the center point and the ID number must be specified.



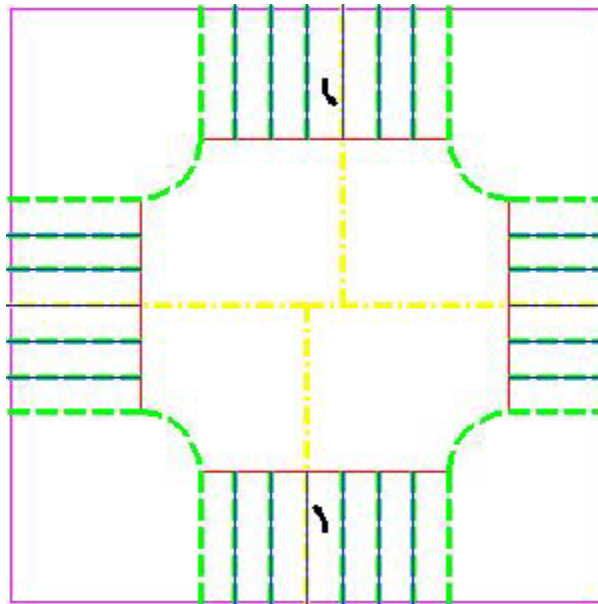
Primitive Command: LOAD FROM - STANDARD - 7X5

Load a standard 4 leg Alternative that has 3 through Inbound Lanes, 1 exclusive left turn Inbound Lane, and 3 Outbound Lanes on the north-south street and 2 through Inbound Lanes, 1 exclusive left turn Inbound Lane, and 2 Outbound Lanes on the east-west street. Coordinates of the center point and the ID number must be specified.



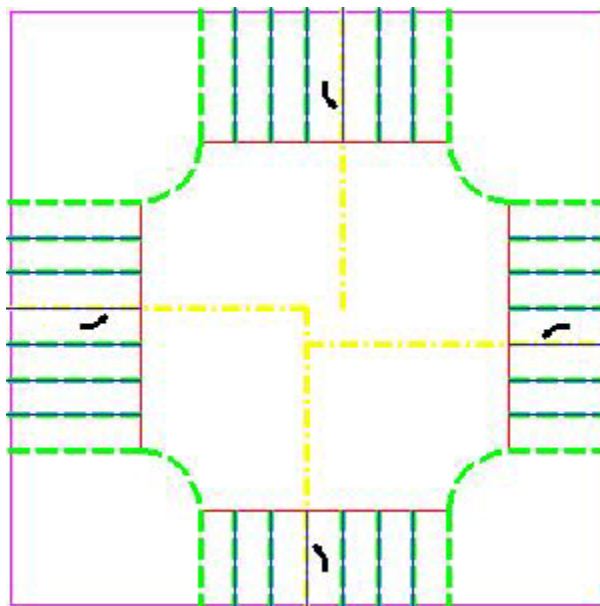
Primitive Command: LOAD FROM - STANDARD - 7X6

Load a standard 4 leg Alternative that has 3 through Inbound Lanes, 1 exclusive left turn Inbound Lane, and 3 Outbound Lanes on the north-south street and 3 through Inbound Lanes and 3 Outbound Lanes on the east-west street. Coordinates of the center point and the ID number must be specified.

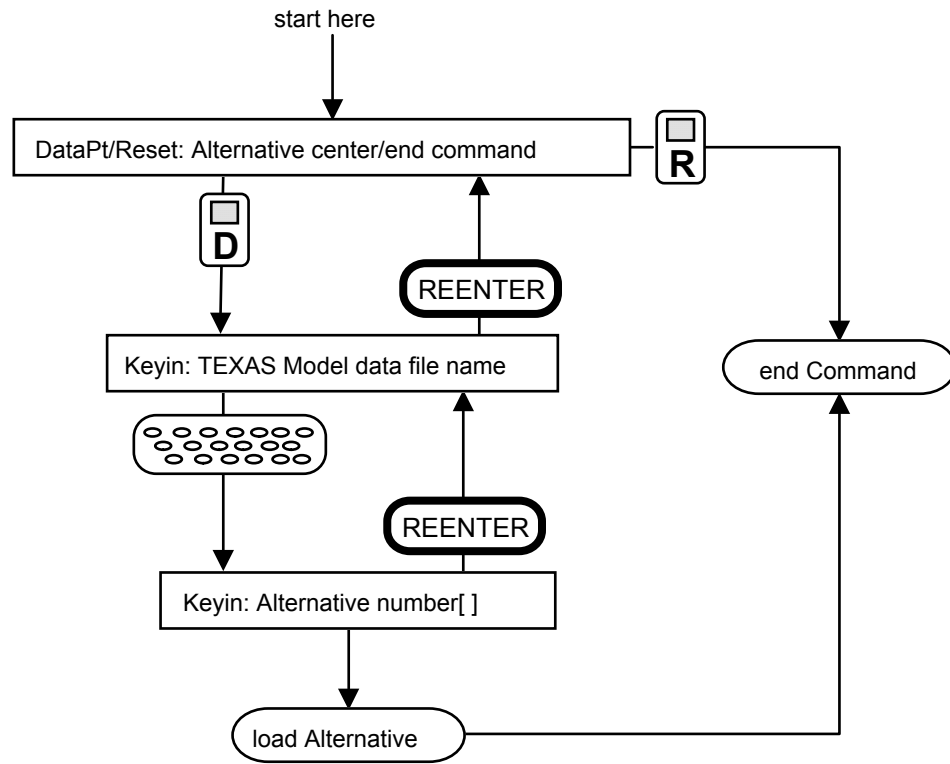


Primitive Command: LOAD FROM - STANDARD - 7X7

Load a standard 4 leg Alternative that has 3 through Inbound Lanes, 1 exclusive left turn Inbound Lane, and 3 Outbound Lanes in each direction. Coordinates of the center point and the ID number must be specified.



Primitive Command: LOAD FROM - TX Mdl file

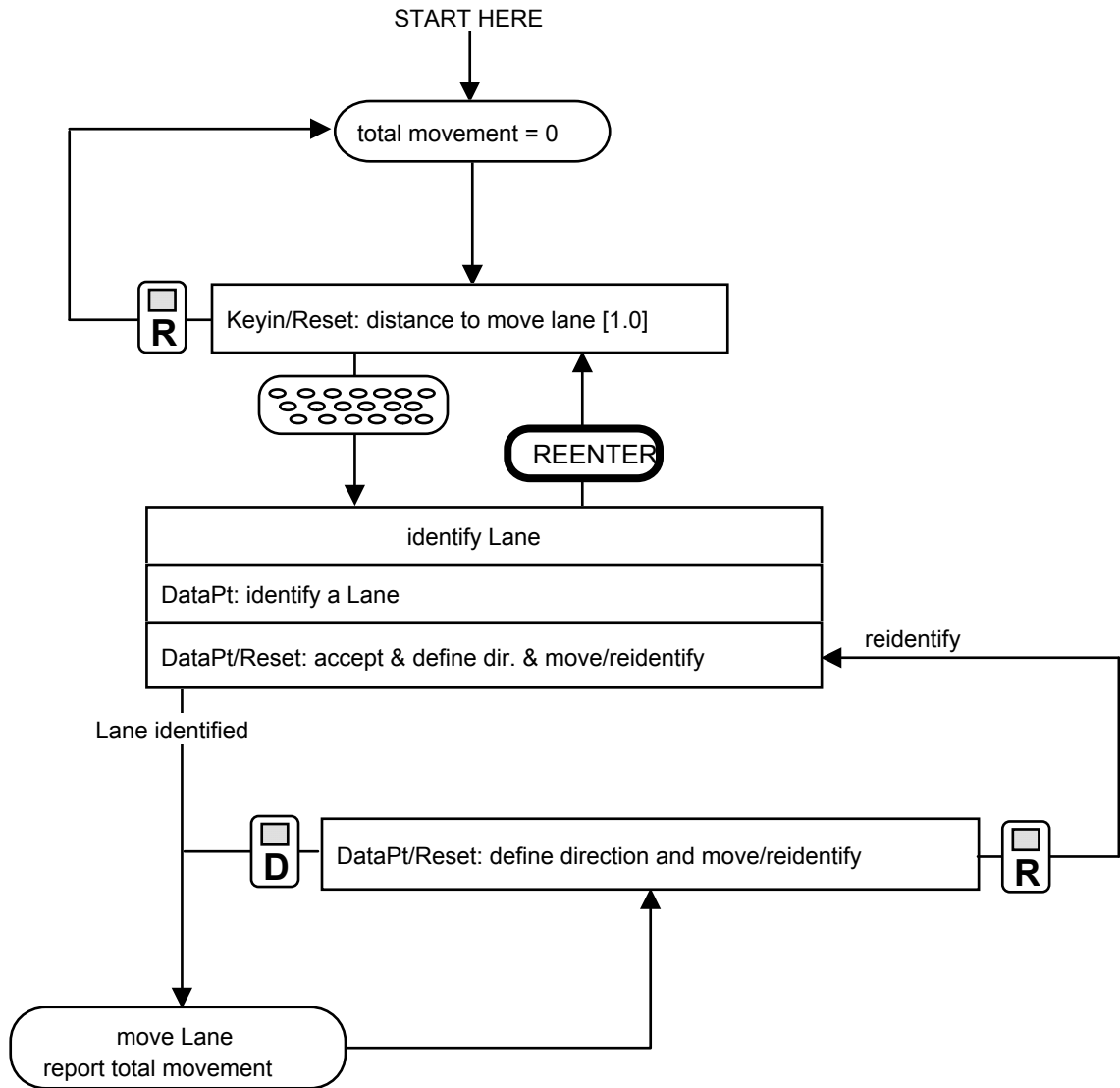


Load data for an Alternative from a file that was written by the TEXAS Model processor called GDVDATA. Specify the coordinates of the center point, data file name and ID number.

Primitive Command: MODIFY - ALTERNATIVE
Not programmed yet.

Primitive Command: MODIFY - INTERSECTION
Not programmed yet.

Primitive Command: MODIFY - LANE - LATERL POSN



Shift a Lane laterally by a specified distance. The direction of movement is determined by the location of the move DataPt with respect to the Leg centerline. The radii of any arcs on the Lane edges will be adjusted by the amount of the shift.

This is the same as **Primitive Command: MOVE - LANE - LATERAL**.

Primitive Command: MODIFY - LANE - LENGTH

Opens the following dialog box and allows the user to move the lane end.

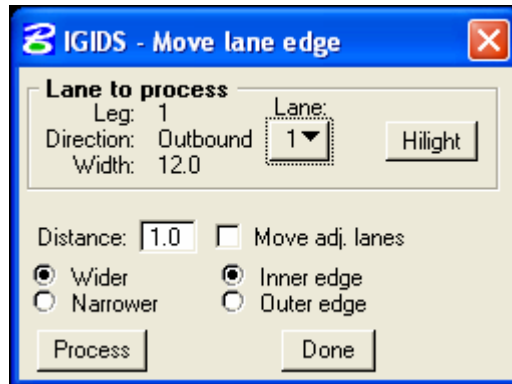


The dialog box is titled "IGIDS - Move lane end" and features a blue header bar with a close button (X) in the top right corner. The main content area is light beige and contains the following elements:

- Lane to process** section:
 - Leg: 1
 - Direction: Outbound
 - Length: 800.0
 - Lane: 1 (dropdown menu)
 - Highlight button
- Distance: 1.00 (text input field)
- Radio button options:
 - ☒ Longer
 - ☐ Shorter
 - ☒ Stopline End
 - ☐ Other End
- Process button (highlighted with a dashed border)
- Done button

Primitive Command: MODIFY - LANE - WIDTH

Opens the following dialog box and allows the user to move the lane edge.

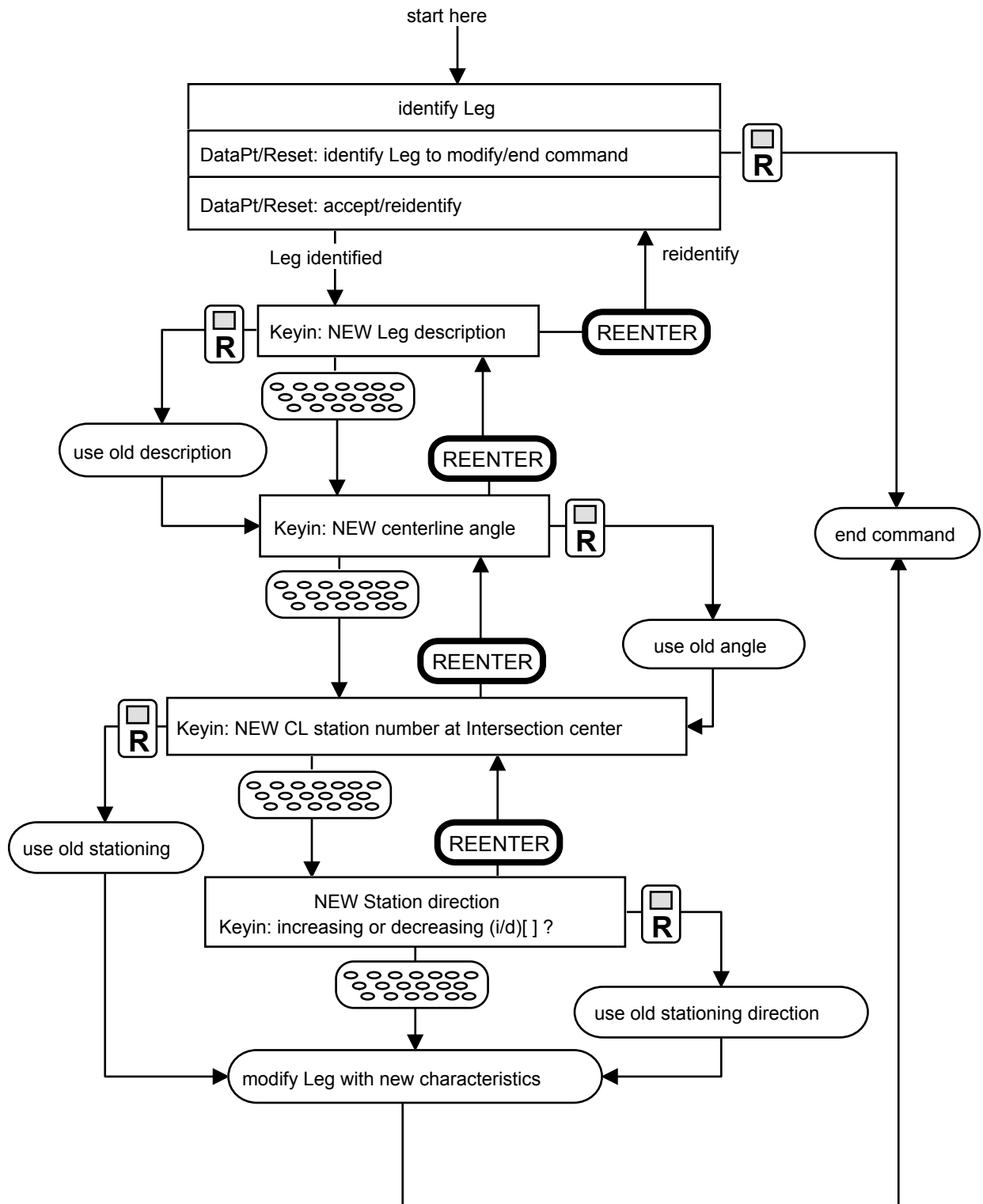


Primitive Command: MODIFY - LANE EDGE - LENGTHEN
Not programmed yet.

Primitive Command: MODIFY - LANE EDGE - SHORTEN
Not programmed yet.

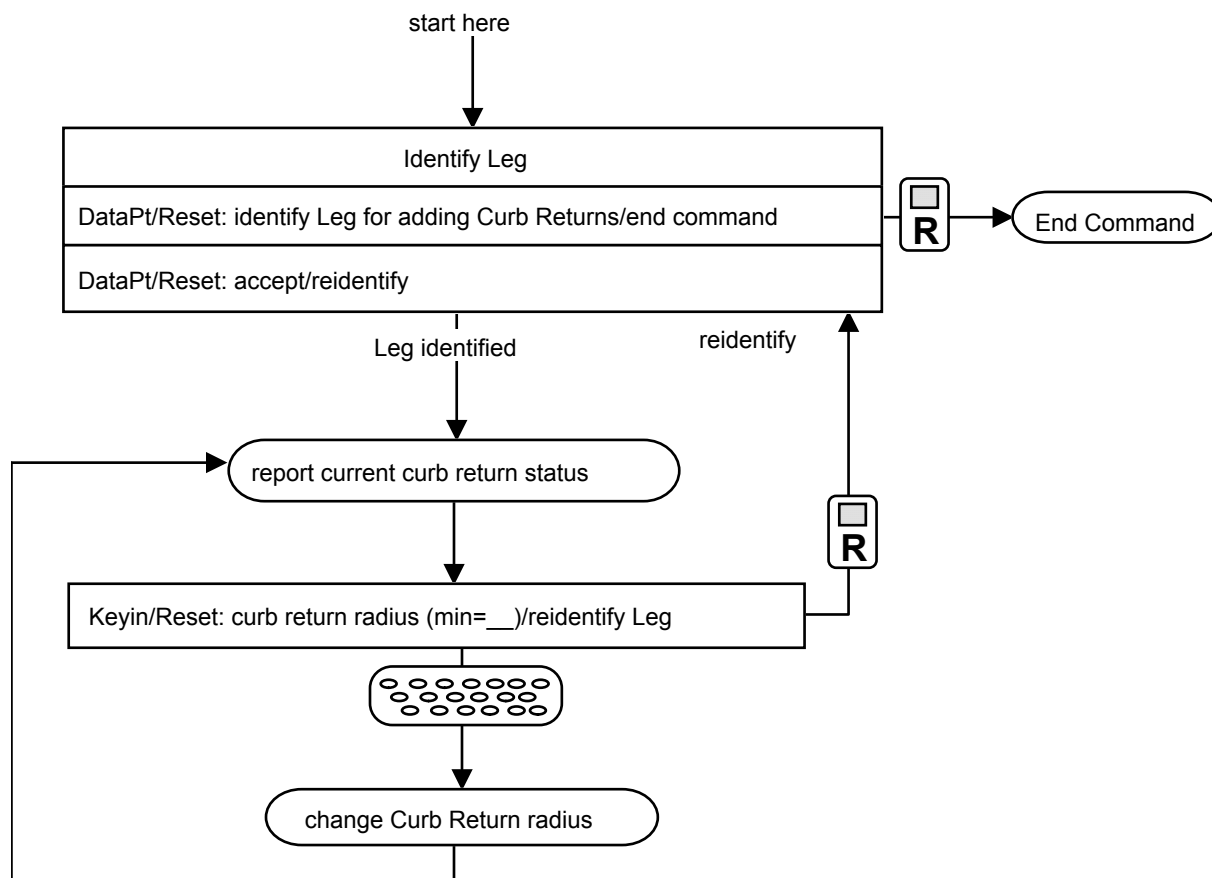
Primitive Command: MODIFY - LANE EDGE - TAPER
Not programmed yet.

Primitive Command: MODIFY - LEG



Change the Leg description, centerline angle, station number at center of intersection and direction of stationing.

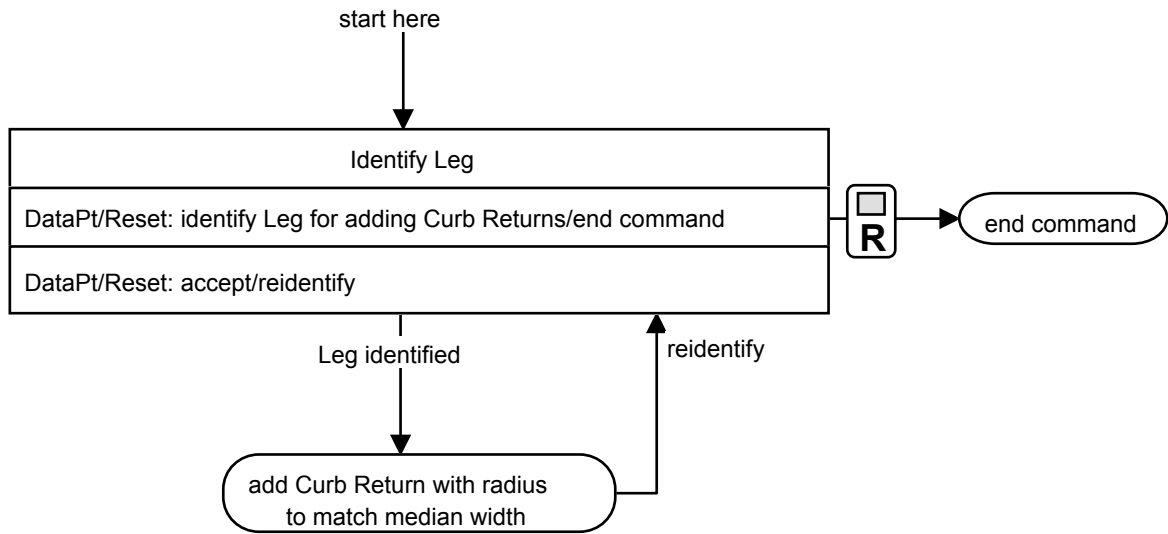
Primitive Command: MODIFY - LEG CURB CR



Add or revise the Curb Return. A keyed in radius is used for the Curb Return radius. This is the Curb Return between the Inbound Lanes of the identified Leg and the Outbound Lanes of the adjacent Leg.

This is the same as **Primitive Command: ADD - LEG CURB CR - BY KEY-IN**

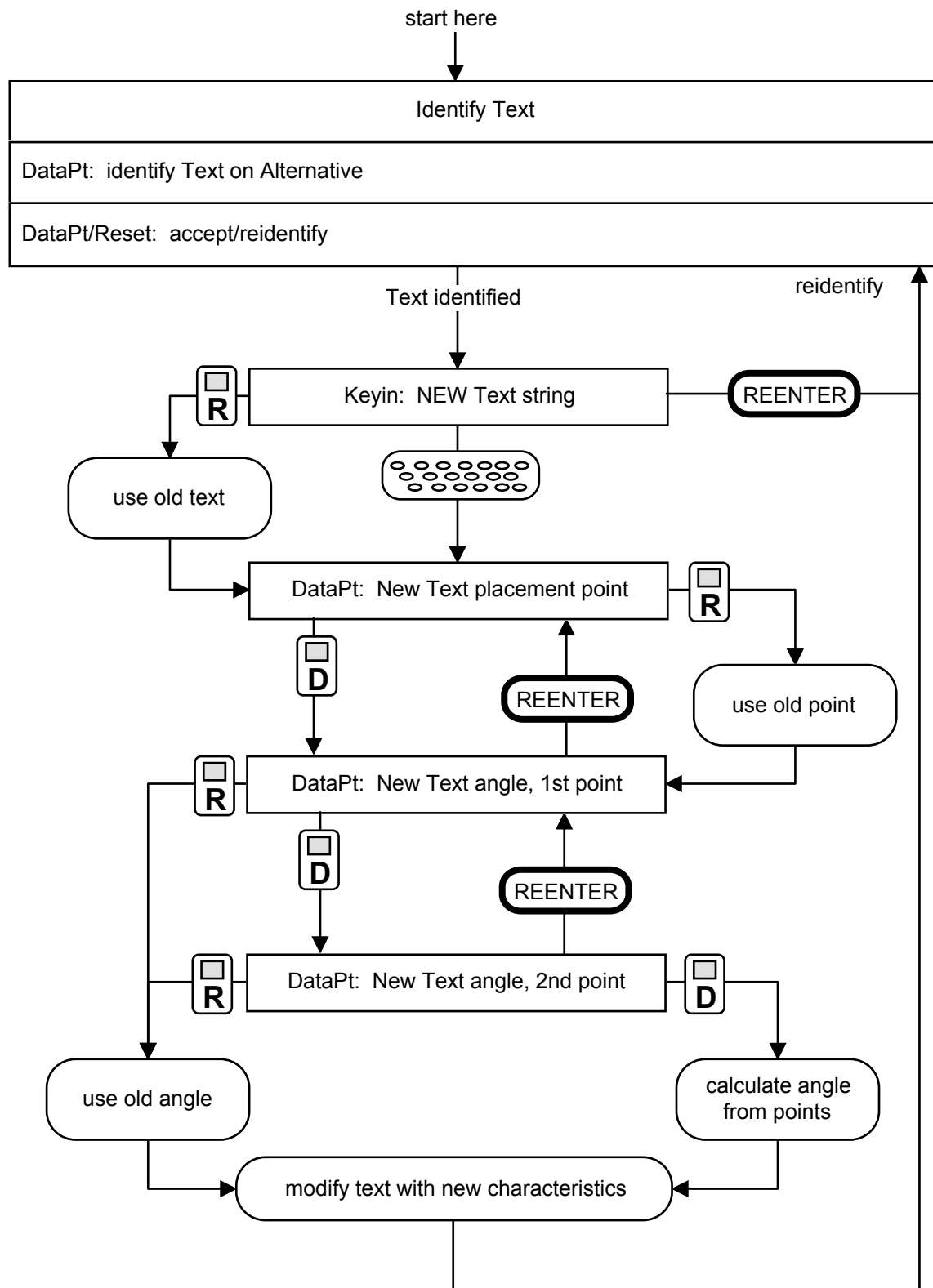
Primitive Command: MODIFY - LEG MEDIAN CR



Add the median Curb Return. The Curb Return radius is automatically set to span the median. This is the Curb Return that closes the median between the Inbound Lanes and Outbound Lanes of the identified Leg.

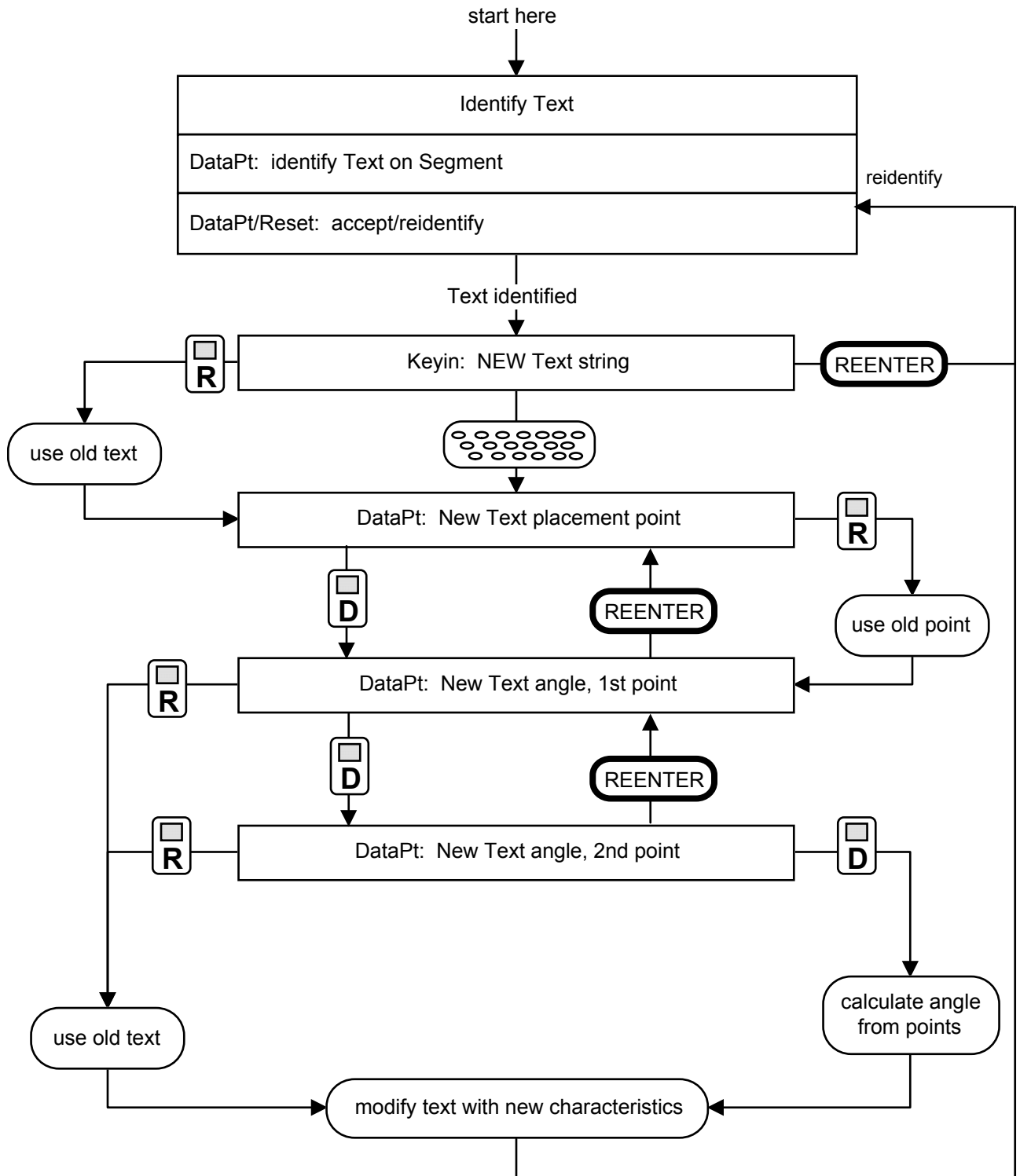
This is the same as **Primitive Command: ADD - LEG MEDIAN CR - BY KEY-IN**

Primitive Command: MODIFY - TEXT ON ALT



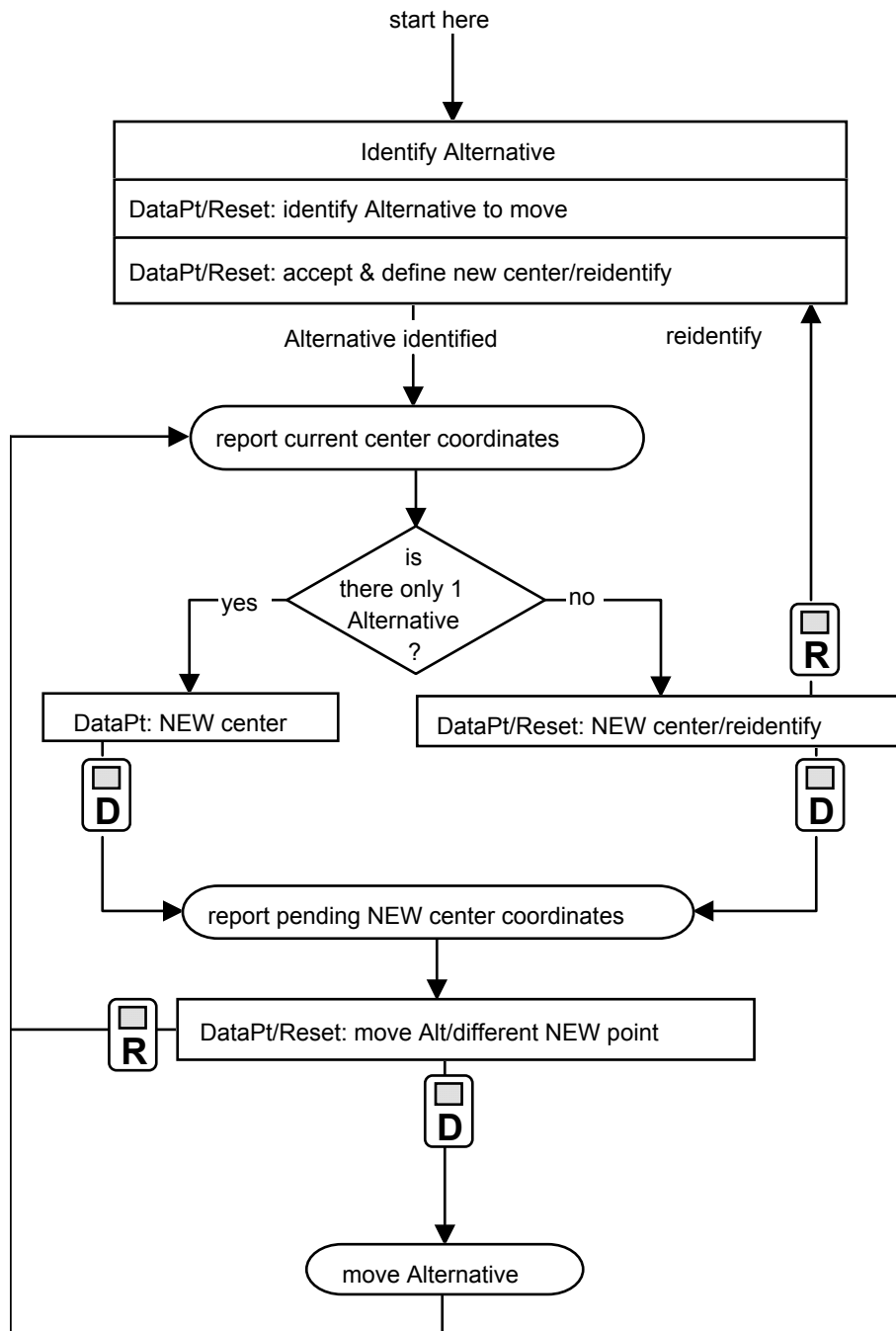
Change the text, location and angle of an existing Text on an Alternative.

Primitive Command: MODIFY - TEXT ON SEG



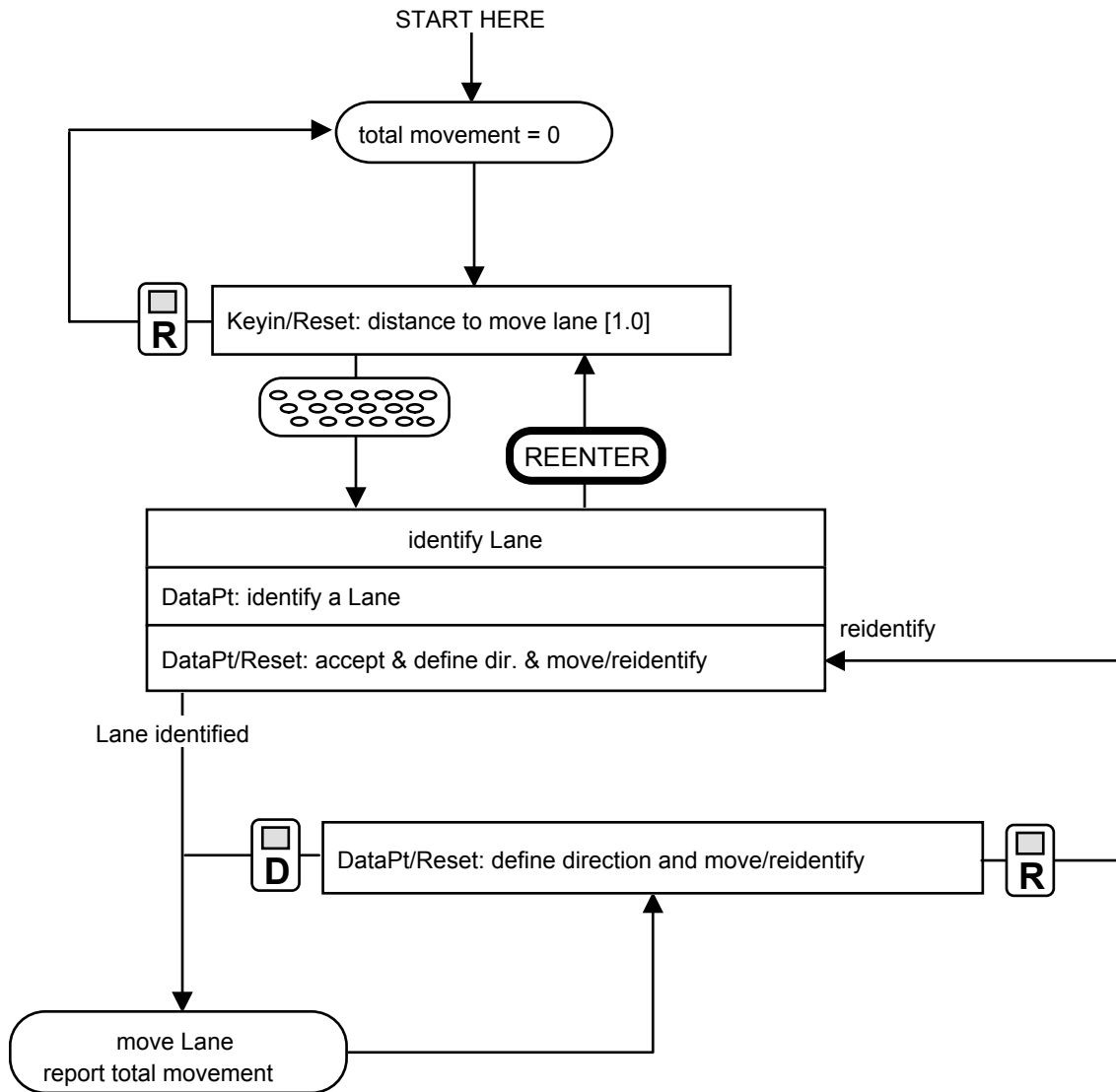
Change the text, location and angle of an existing Text on a Seg.

Primitive Command: MOVE - ALTERNATIVE



Move an Alternative by specifying a new center point.

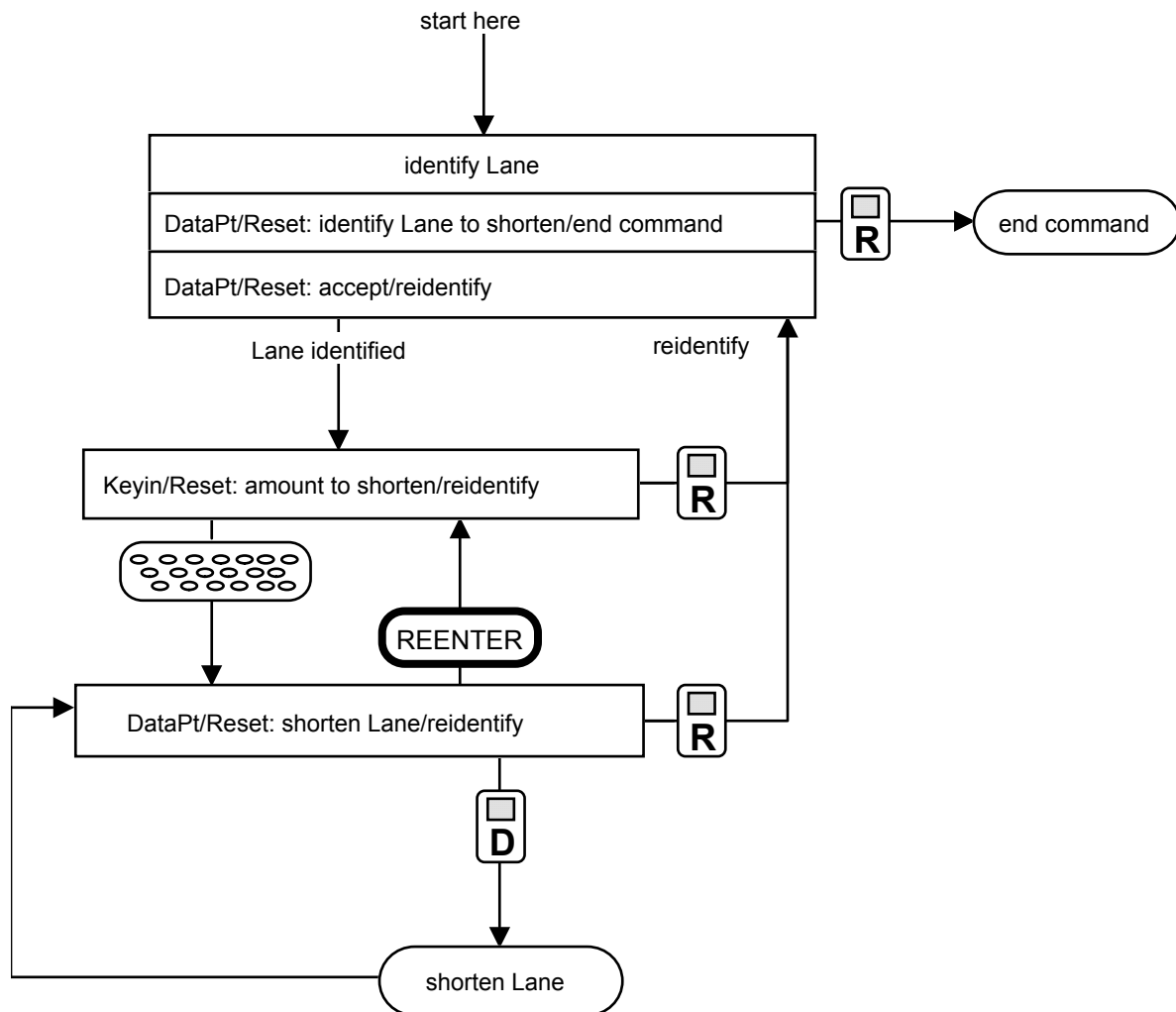
Primitive Command: MOVE - LANE - LATERAL



Shift a Lane laterally by a specified distance. The direction of movement is determined by the location of the move DataPt with respect to the Leg centerline. The radii of any arcs on the Lane edges will be adjusted by the amount of the shift.

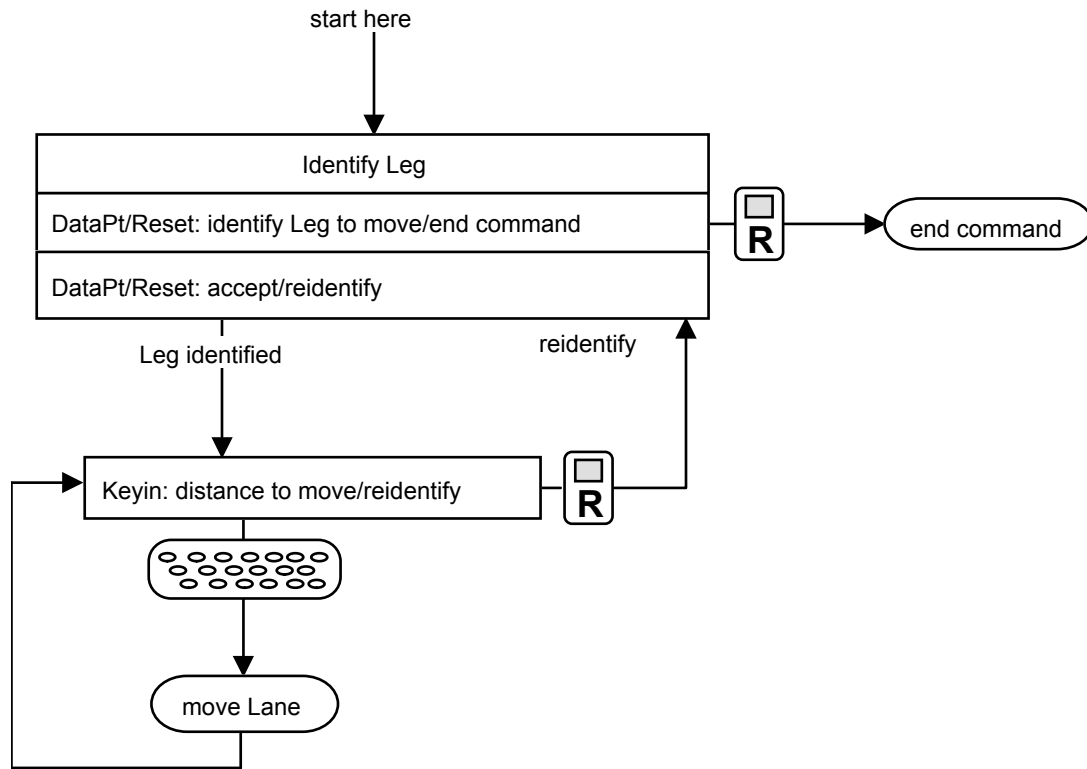
This is the same as **Primitive Command: MODIFY - LANE - LATERL POSN.**

Primitive Command: MOVE - LANE - LONGITUDINAL



Shorten the end of a Lane that is nearest the intersection center.

Primitive Command: MOVE - LEG - LATERAL



Move a Leg laterally by a specified distance. The direction of movement is determined by the arithmetic sign of the distance. Plus will move to the right when facing in the direction of inbound traffic.

Primitive Command: MOVE - LEG - LONGITUDINAL

identify Leg to move

keyin: not allowed

datapt: select object

reset: end command

reenter: not allowed

Keyin: distance to move

keyin: double; - is toward intersection center; + is away from intersection center

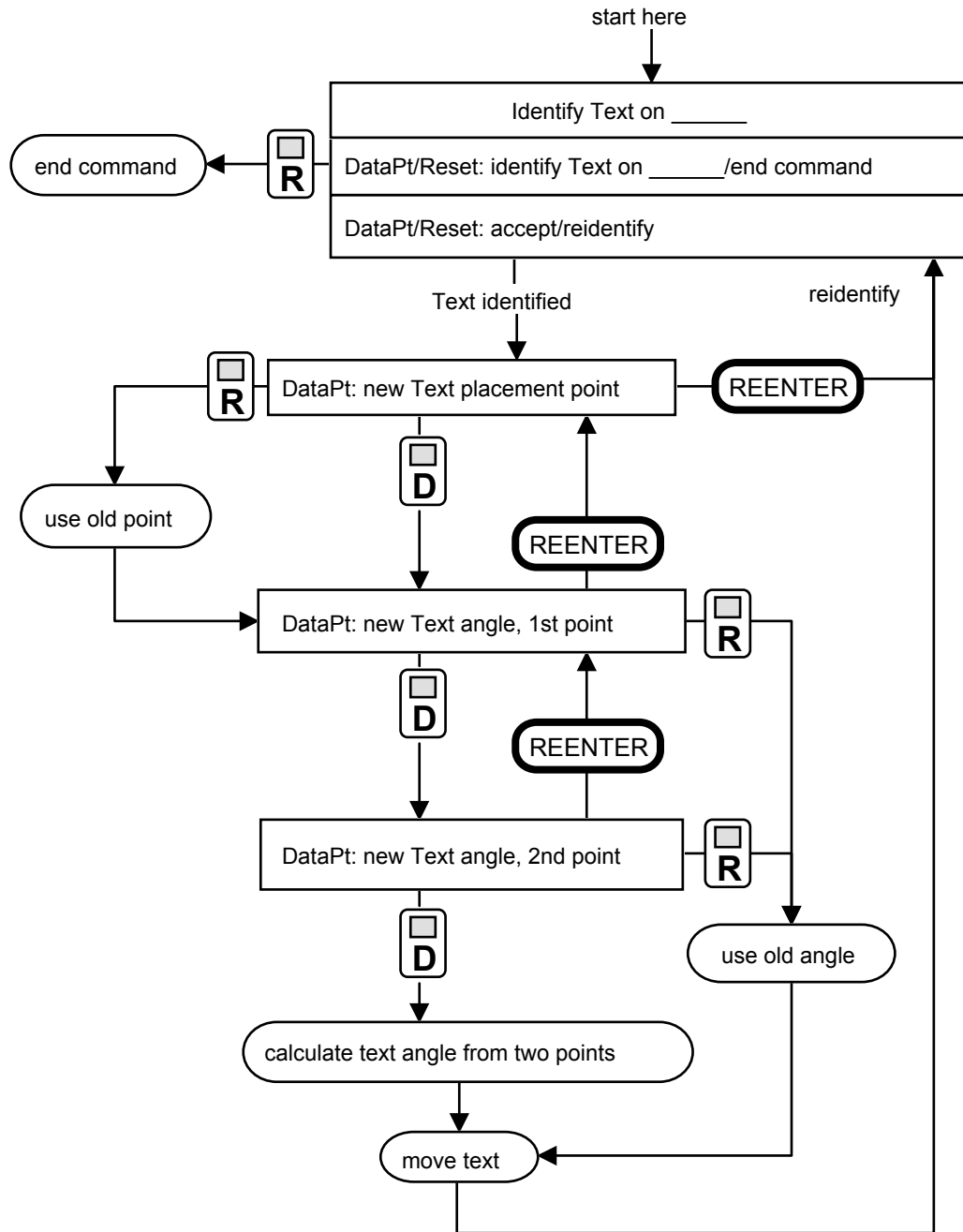
datapt: not allowed

reset: end command

reenter: not allowed

move leg longitudinal

Primitive Command: MOVE - TEXT



Use this diagram for the 2 **MOVE - TEXT** commands that follow.

Primitive Command: MOVE - TEXT ON ALT

Move an existing Text on an Alternative. The rotation angle may also be changed.

Primitive Command: MOVE - TEXT ON SEG

Move an existing Text on a Seg. The rotation angle may also be changed.

Primitive Command: MOVE - TRAF CONTRL

identify Traffic Control/end command

keyin: not allowed

datapt: select object

reset: end command

reenter: not allowed

DataPt: NEW Traffic Control placement point

keyin: precision keyin allowed

datapt: new placement point

reset: use current placement point for new placement point

reenter: backup to identify Traffic Control/end command

DataPt: New Traffic Control angle, 1st point

keyin: precision keyin allowed

datapt: 1st point for angle

reset: use current angle for new angle

reenter: backup to DataPt: NEW Traffic Control placement point

if not reset on last command then DataPt: New Traffic Control angle, 2nd point

keyin: precision keyin allowed

datapt: 2nd point for angle

reset: use current angle for new angle

reenter: backup to DataPt: NEW Traffic Control placement point

selected traffic control symbol is moved and rotated.

This is the same as **Primitive Command: TOOLS - Traffic - Controller - MOVE**, **Primitive Command: TOOLS - Traffic - Sign - MOVE**, and **Primitive Command: TOOLS - Traffic - Signal Face - MOVE**.

Transient Command: No

In reply to a prompt requesting a "yes" or "no" response, send "no" to IGIDS. This is the same as entering "no" through the keyboard.

Transient Command: Noun-Verb or Verb-Noun

This command toggles between the two methods of command processing. The **Noun-Verb** method always uses the selected IGIDS Object as the default choice when identifying an Object for processing. The **Verb-Noun** method always prompts the user to identify an Object for processing.

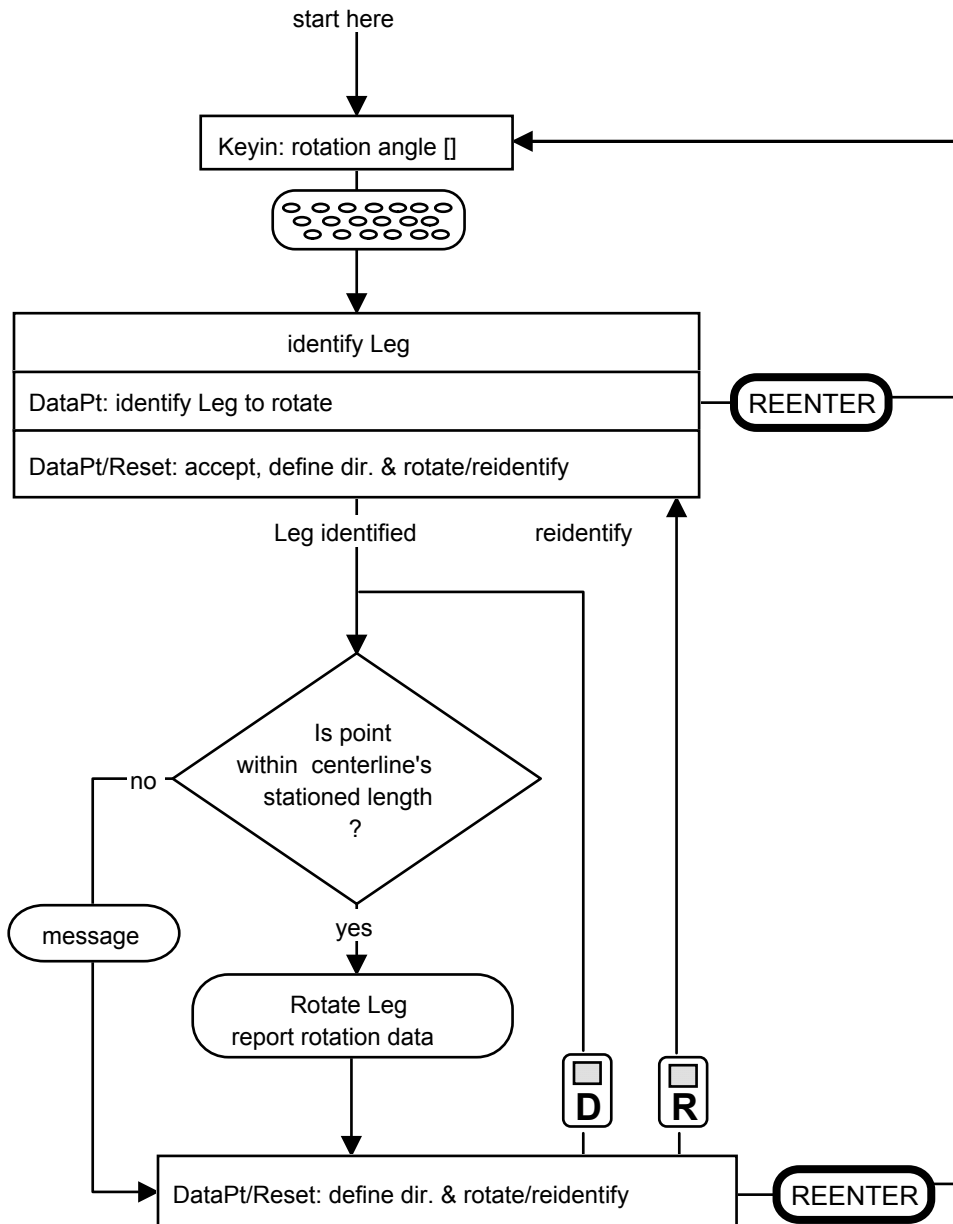
Transient Command: Reenter Data

In an IGIDS command, move backward in the processing sequence to where IGIDS most recently prompted for a Keyin of data and reprompt for the data.

Primitive Command: ROTATE - ALTERNATIVE

Not programmed yet.

Primitive Command: ROTATE - LEG



Rotate a Leg through a specified angle. The direction of rotation is determined by the direction of the rotate DataPt with respect to the Leg centerline.

Primitive Command: ROTATE - TEXT ON ALT
Not programmed yet.

Primitive Command: ROTATE - TEXT ON SEG
Not programmed yet.

Primitive Command: ROTATE - TRAF CONTRL

Keyin: rotation angle [1.0]

keyin: double; - is CW; + is CCW; if fabs < 0.0001 then error message and prompt again

datapt: not allowed

reset: prompt again

reenter: not allowed

identify Traffic Control to rotate

accept & rotate

Traffic Control

keyin: not allowed

datapt: identify, accept object, and rotate

reset: not allowed

reenter: backup to Keyin: rotation angle

DataPt/Reset: rotate/reidentify

keyin: not allowed

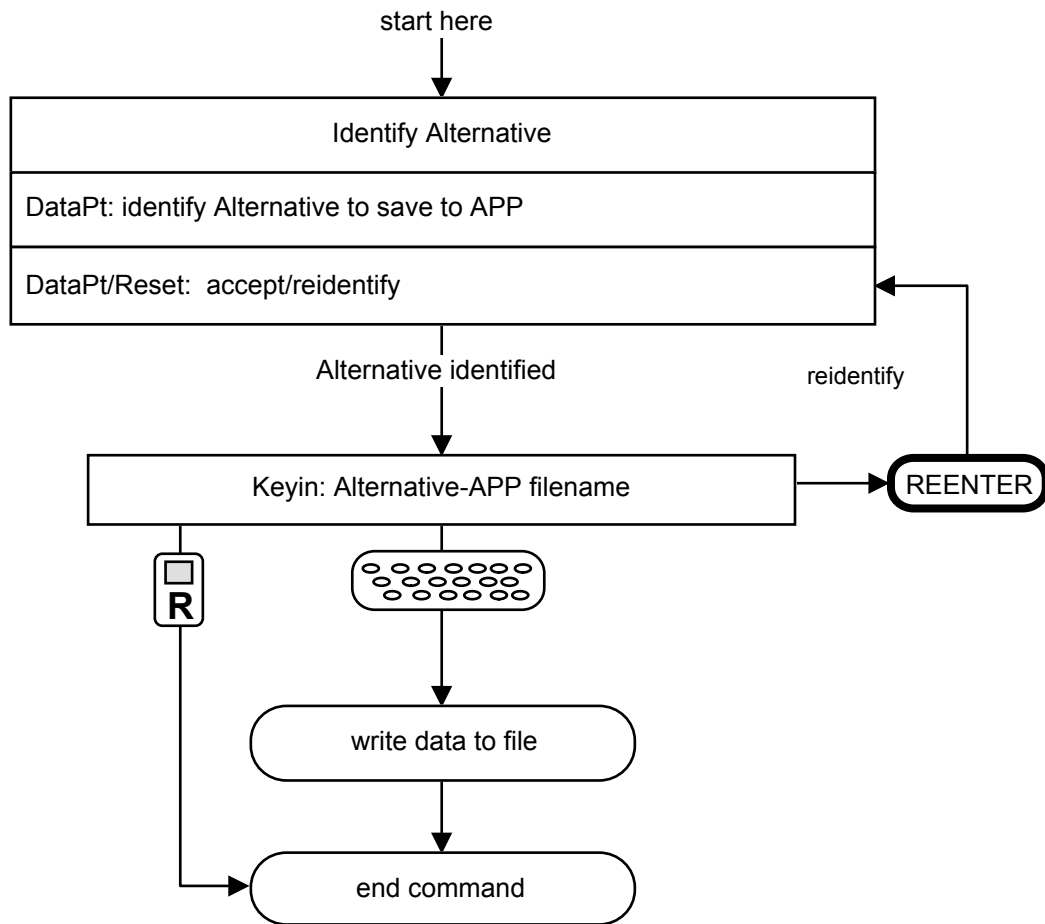
datapt: repeat rotate and prompt again

reset: backup to identify Traffic Control to rotate

reenter: backup to Keyin: rotation angle [1.0]

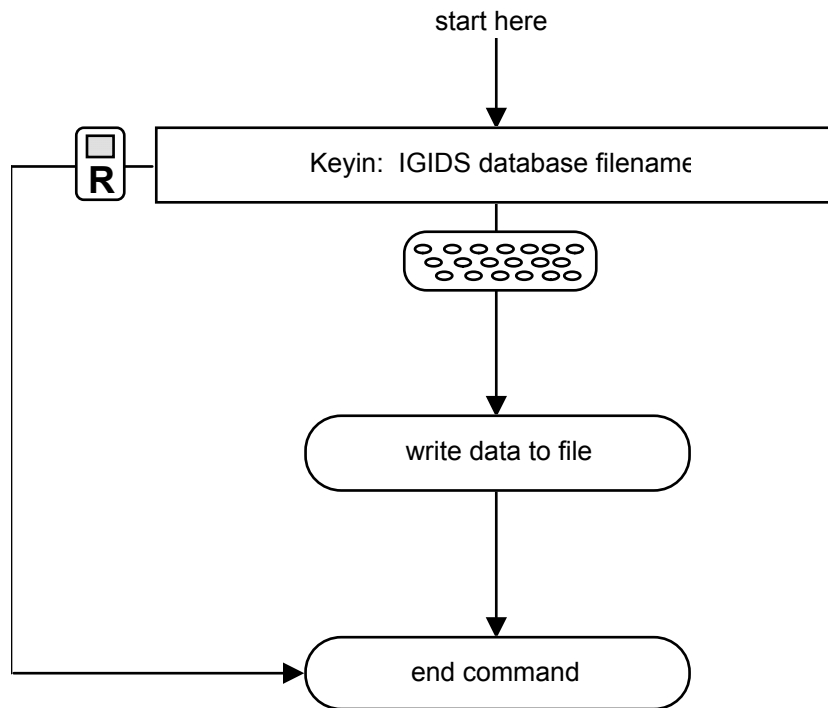
selected traffic control symbol is rotated.

Primitive Command: SAVE TO - AutoPlanPrep



Save data from one Alternative into a file that can be read by TX-DOT's Automatic Plan Preparation software.

Primitive Command: SAVE TO - Data Base



Save data from all Alternatives into a file that can be read by IGIDS. This will save all IGIDS data from the current session. Data saved by this command can be read by **Primitive Command: LOAD FROM - DATABASE**.

Primitive Command: SAVE TO - PASSER II 90

identify an Alternative for PASSER II-90 analysis

Alternative for PASSER II-90 analysis

keyin: not allowed

datapt: identify and accept alternative; if not 4-leg intersection then if more than 1 alternative then error message and prompt again else end command; if number of phases is invalid then if more than 1 alternative then error message and prompt again else end command

reset: not allowed

reenter: not allowed

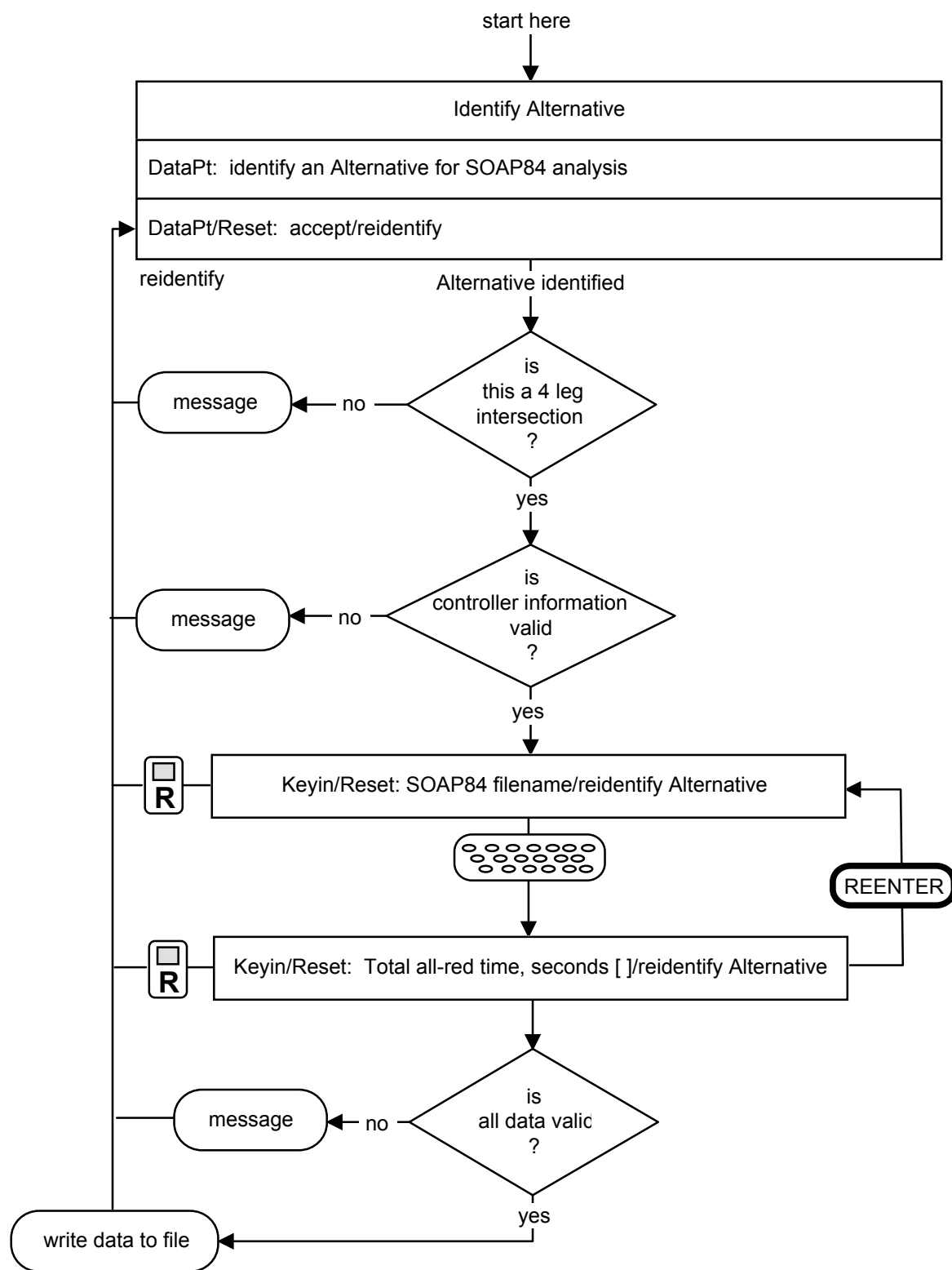
IGIDS PASSER II-90 file name; default = P290DATA

OK: create file

cancel: end command

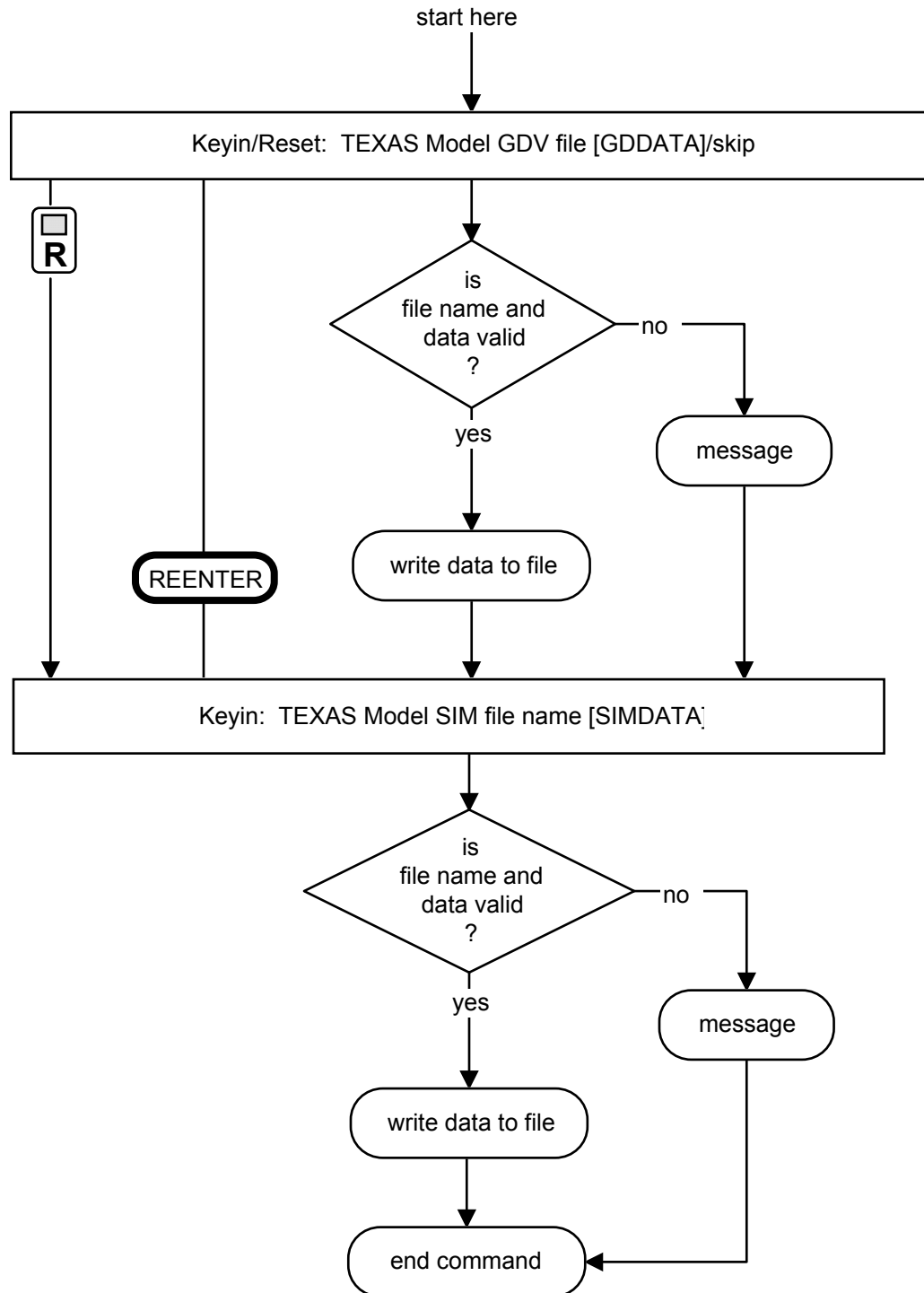
Save data from one Alternative into a file that can be read by the PASSER II 90 software.

Primitive Command: SAVE TO - SOAP



Save data from one Alternative into a file that can be read by the Signal Operations Analysis Package software.

Primitive Command: SAVE TO - TX Mdl file



Save data from the currently selected Alternative into files that can be read by the TEXAS Model for Intersection Traffic. The first file will be readable by the TEXAS Model processor called GDVDATA. The second file will be readable by the TEXAS Model processor called SIMDATA.

Transient Command: SELECT - ALTERNATE - BY DATA PT

identify Alternative/continue

accept Alternative/reidentify

Alternative

keyin: not allowed

datapt: identify and accept alternative

reset: not allowed

reenter: not allowed

Make the identified Alternative the selected Alternative

Transient Command: SELECT - ALTERNATE - BY ID

Specify the ID number of the Alternative to become the selected Alternative

Transient Command: SELECT - ALTERNATE - CURRENT

Make the current Alternative the selected Alternative

This is the same as **Transient Command: HILITE - CURRENT ALT**

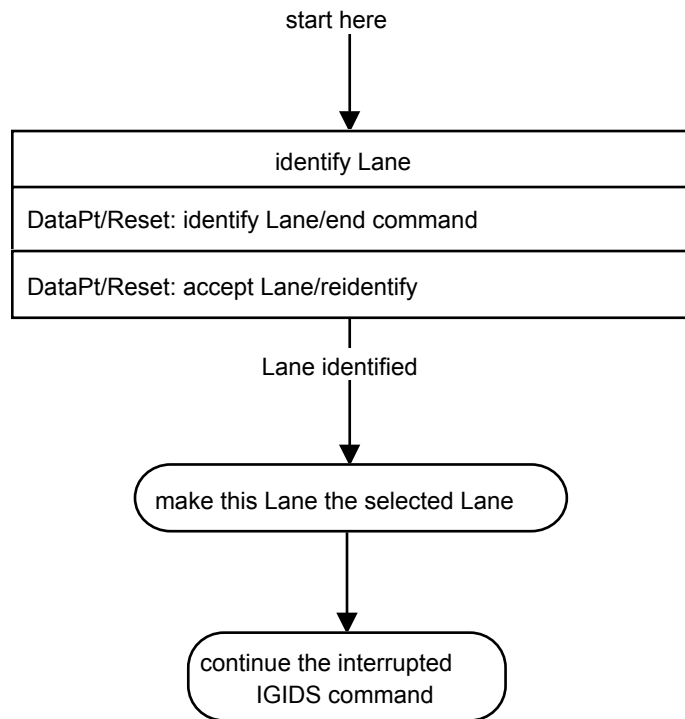
Transient Command: SELECT - ALTERNATIVE - NEXT

Make the next Alternative in the selection list the selected Alternative

Transient Command: SELECT - ALTERNATIVE - PREVIOUS

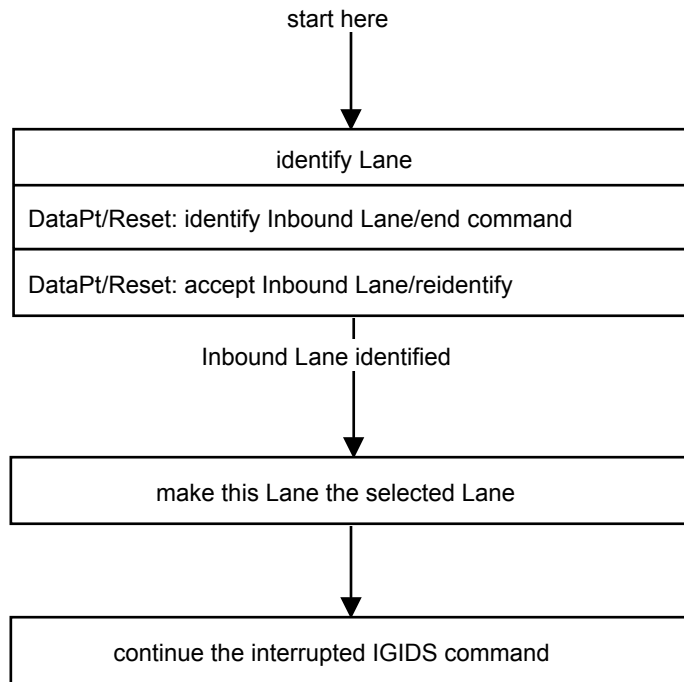
Make the previous Alternative in the selection list the selected Alternative

Temporary Command: SELECT - LANE - BY DATA PT



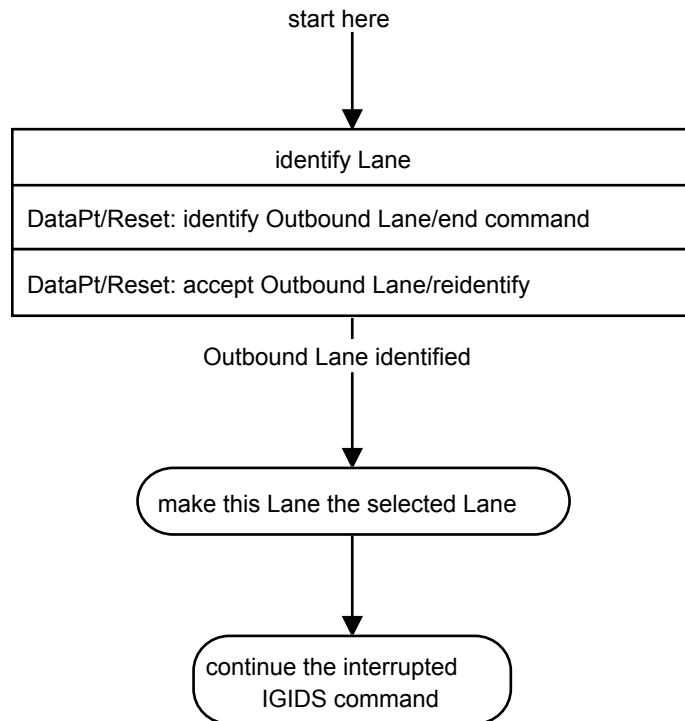
Identify a Lane to be the selected Lane.

Temporary Command: SELECT - LANE - BY DATA PT - INBOUND



Identify an Inbound Lane to be the selected Lane.

Temporary Command: SELECT - LANE - BY DATA PT - OUTBOUND



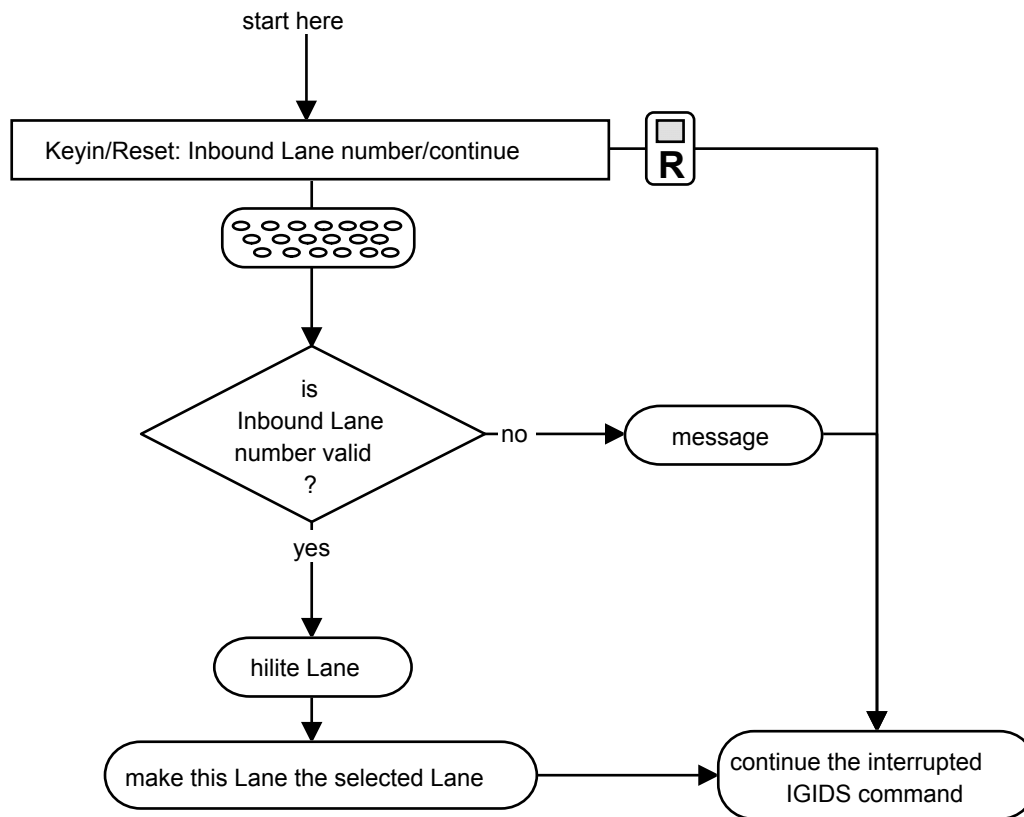
Identify an Outbound Lane to be the selected Lane.

Transient Command: SELECT - LANE - CURRENT

Make the current Lane the selected Lane.

This is the same as **Transient Command: HILITE - CURRENT LANE - ALL.**

Temporary Command: SELECT - LANE - INBOUND ID



Specify the ID number of an Inbound Lane on the selected Leg to be the selected Lane.

Transient Command: SELECT - LANE - NEXT

Make the Lane with the next higher ID number the selected Lane. If the currently selected lane has the highest ID number, make the opposite direction Lane with the ID number 1 the selected Lane.

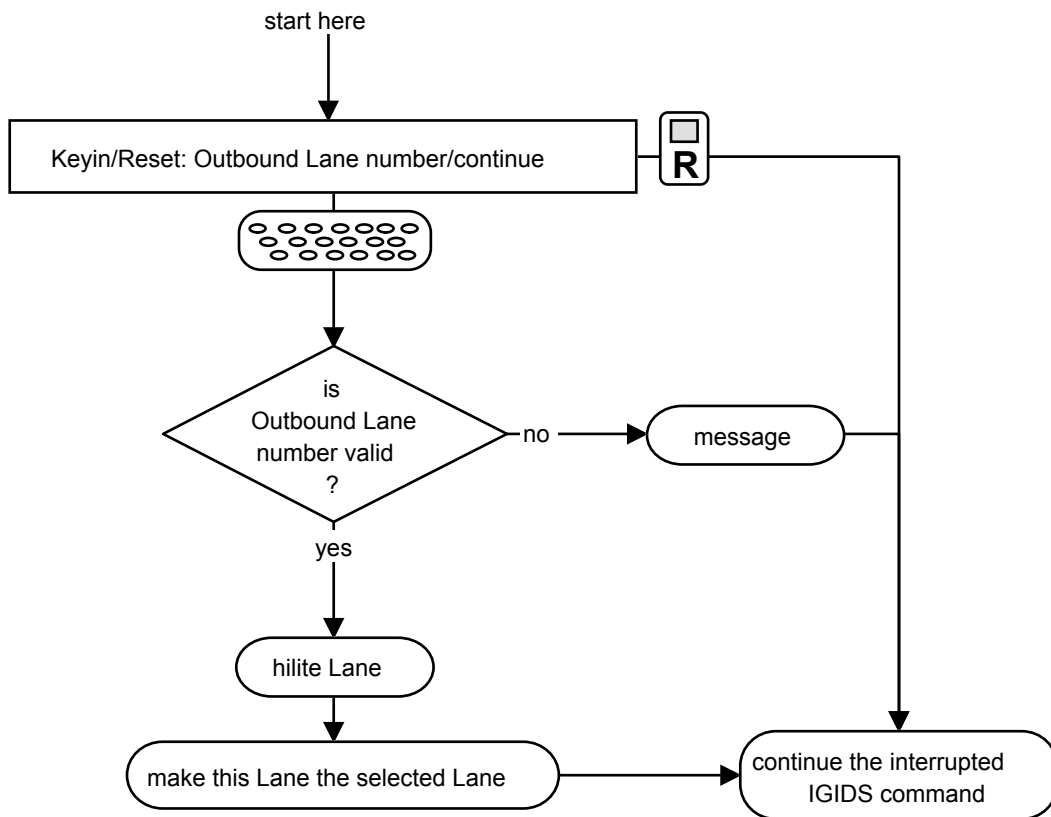
Transient Command: SELECT - LANE - NEXT - INBOUND

Make the Inbound Lane with the next higher ID number the selected Lane

Transient Command: SELECT - LANE - NEXT - OUTBOUND

Make the Outbound Lane with the next higher ID number the selected Lane

Temporary Command: SELECT - LANE - OUTBOUND ID



Specify the ID number of an Outbound Lane on the selected Leg to be the selected Lane.

Transient Command: SELECT - LANE - PREVIOUS

Make the Lane with the next lower ID number the selected Lane. If the currently selected lane has ID number 1, make the opposite direction Lane with the highest ID number the selected Lane.

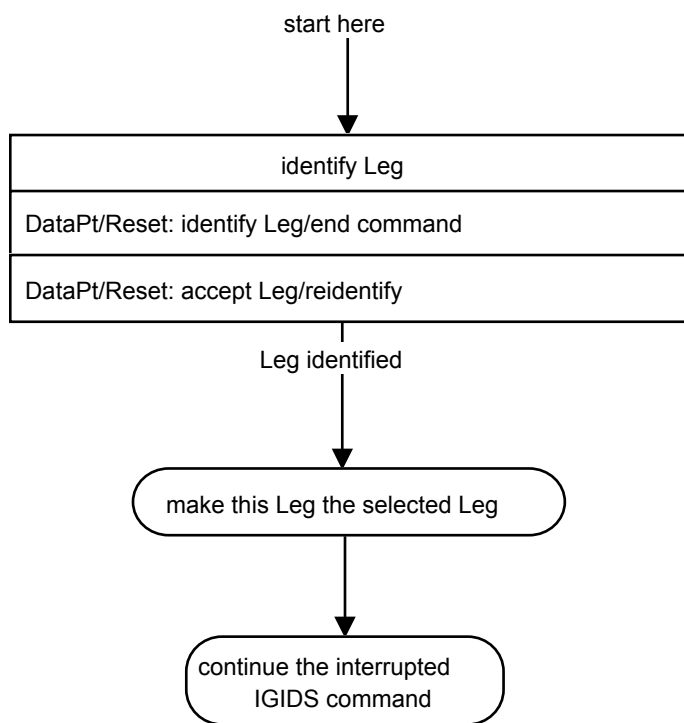
Transient Command: SELECT - LANE - PREVIOUS - INBOUND

Make the Inbound Lane with the next lower ID number the selected Lane

Transient Command: SELECT - LANE - PREVIOUS - OUTBOUND

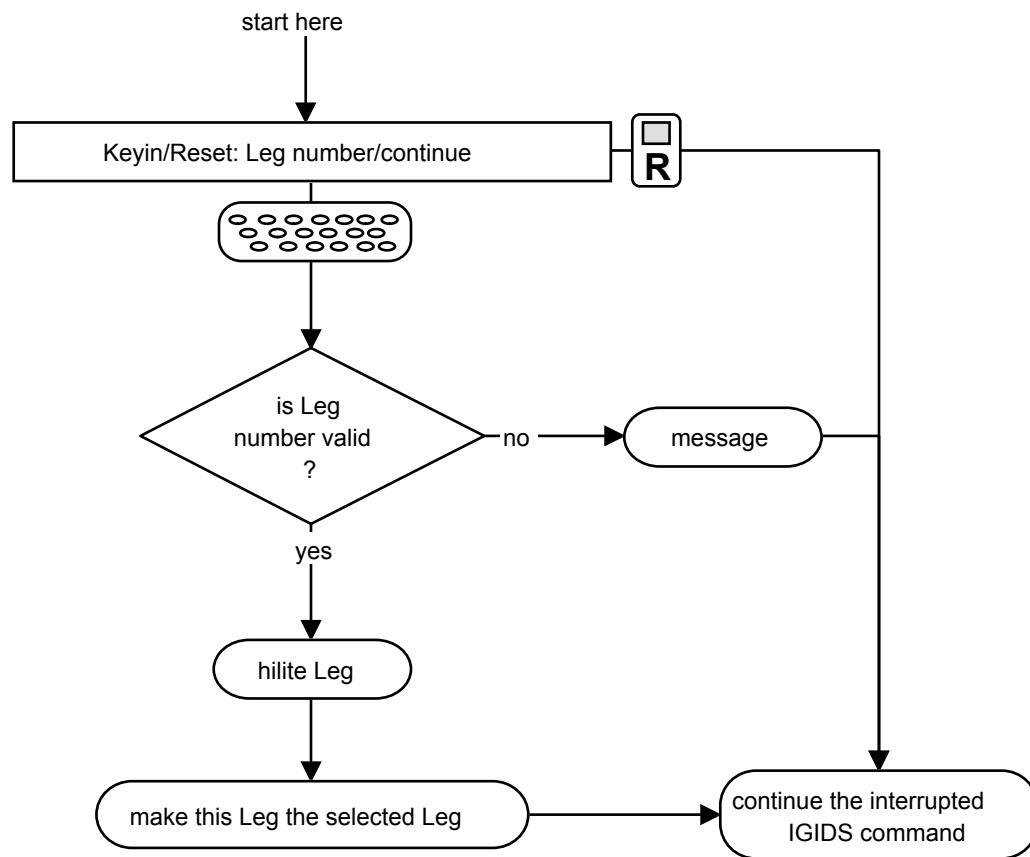
Make the Outbound Lane with the next lower ID number the selected Lane

Temporary Command: SELECT - LEG - BY DATA PT



Identify a Leg to be the selected Leg.

Temporary Command: SELECT - LEG - BY ID



Specify the ID number of a Leg in the selected Alternative to be the selected Leg.

Transient Command: SELECT - LEG- CURRENT

Make the current Leg the selected Leg.

This is the same as **Transient Command: HILITE - CURRENT LEG - ALL.**

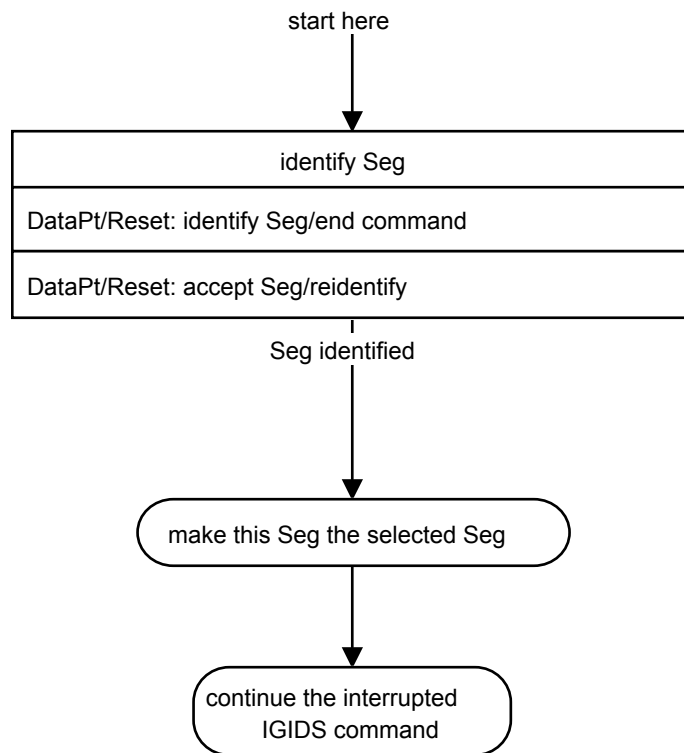
Transient Command: SELECT - LEG - NEXT

Make the Leg that is nearest to the selected Leg in a clockwise direction the selected Leg.

Transient Command: SELECT - LEG - PREVIOUS

Make the Leg that is nearest to the selected Leg in a counterclockwise direction the selected Leg.

Temporary Command: SELECT - SEGMENT - BY DATA PT



Identify the Seg to become the selected Seg

Temporary Command: SELECT - SEGMENT - BY ID

Specify the ID number of the Segment to become the selected Segment.

Transient Command: SELECT - SEGMENT - CURRENT

Hilite the selected Seg

This is the same as **Transient Command: HILITE - CURRENT SEG.**

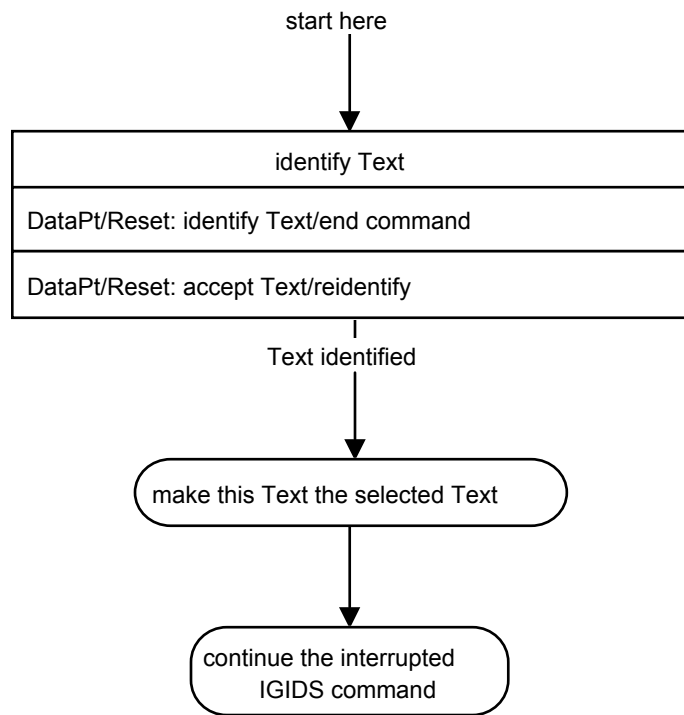
Transient Command: SELECT - SEGMENT - NEXT

Make the next Seg in the selection list the selected Seg

Transient Command: SELECT - SEGMENT - PREVIOUS

Make the previous Seg in the selection list the selected Seg

Temporary Command: SELECT - TEXT - BY DATA PT



Identify the Text to become the selected Text

Temporary Command: SELECT - TEXT - BY ID

Specify the ID number of the Text to become the selected Text

Transient Command: SELECT - TEXT - CURRENT

Hilite the selected Text

This is the same as **Transient Command: HILITE - CURRENT TEXT.**

Transient Command: SELECT - TEXT - NEXT

Make the next Text in the selection list the selected Text

Transient Command: SELECT - TEXT - PREVIOUS

Make the previous Text in the selection list the selected Text.

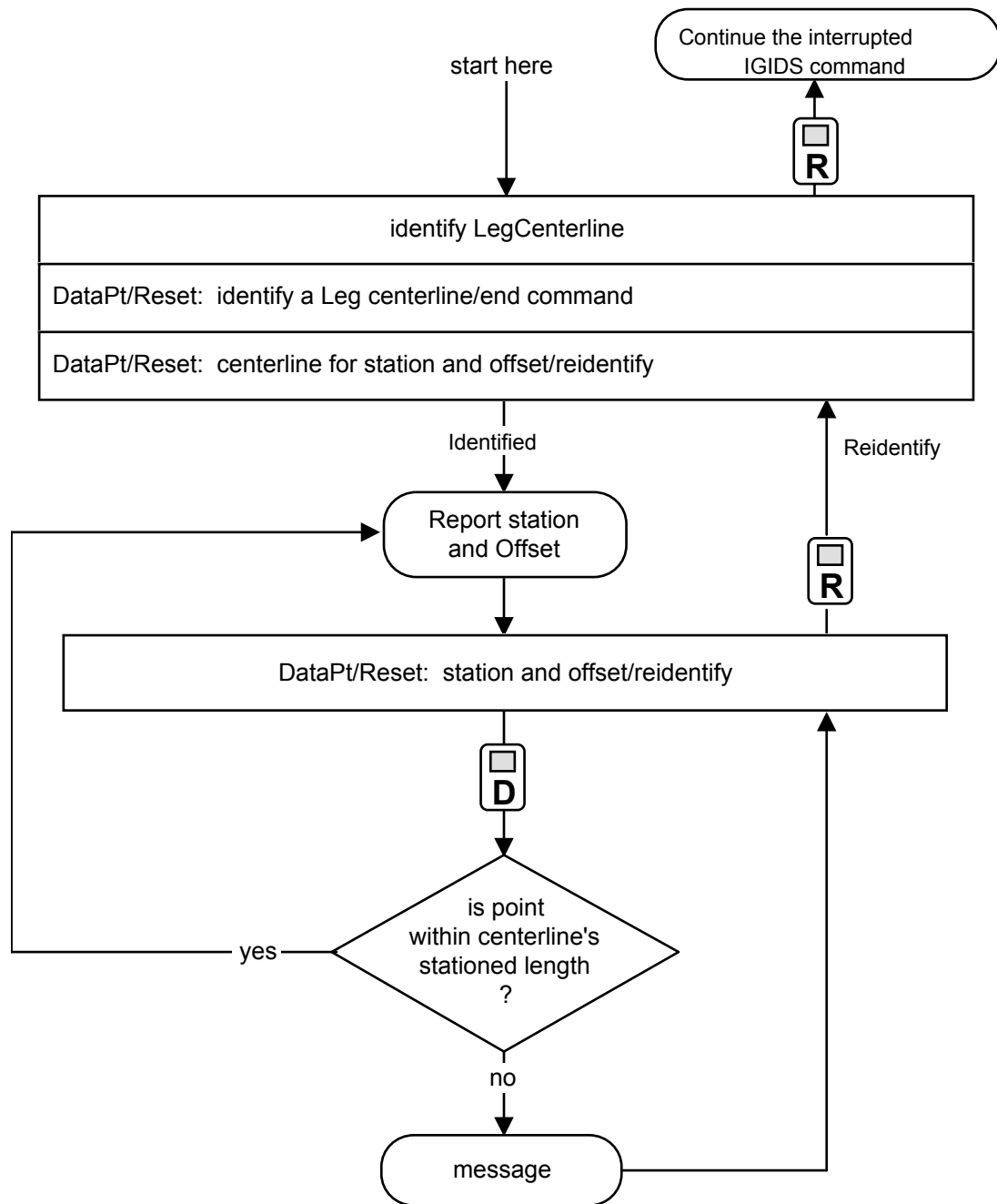
Transient Command: SHOW INFO - FULL

Show a more complete description of the currently selected objects in a window.

Transient Command: SHOW INFO - SHORT

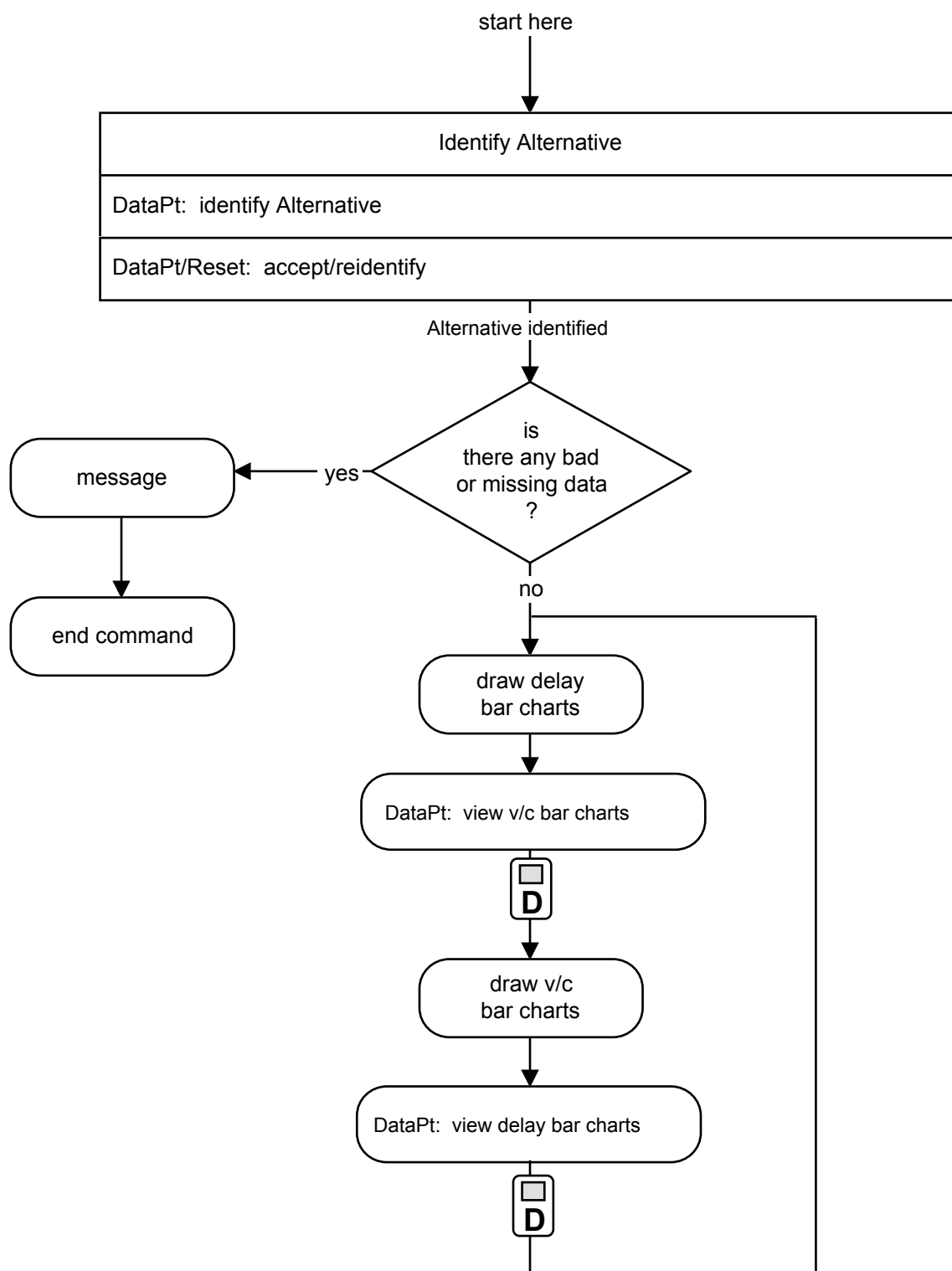
Provide a single line of information about the currently selected Objects. This line will list the current ID number of the selected Alternative, Leg and Lane.

Temporary Command: Sta/Offset



Report the station and offset of a point, based on the stationing of an identified Leg centerline.

Primitive Command: TOOLS - HighCapMan - Chapter 9



A partial implementation of the procedures described in Chapter 9 of the Highway Capacity Manual. Determination of the critical lane groups for the intersection is not implemented

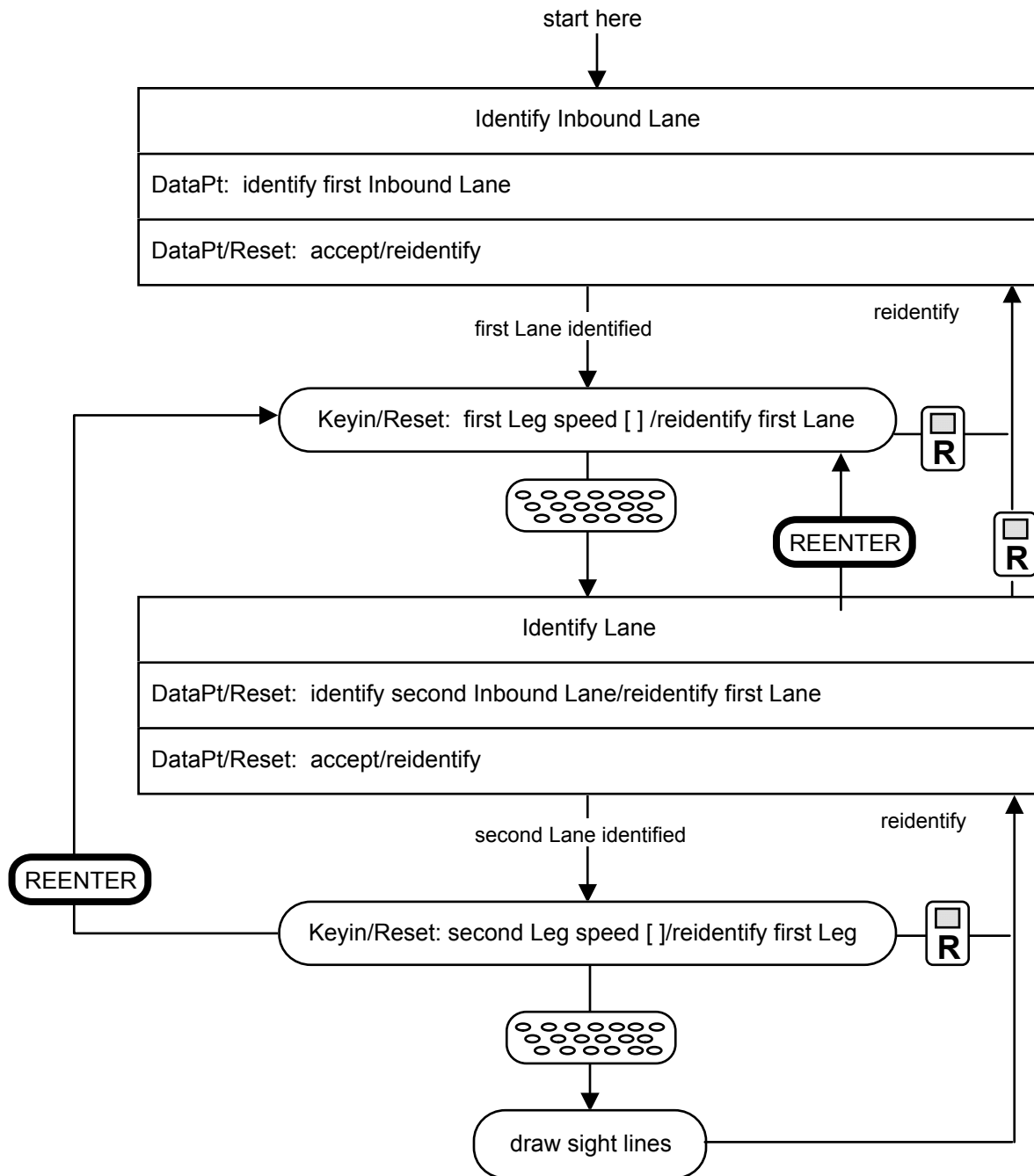
Transient Command: TOOLS - HighCapMan - Del Graphics

Delete all of the Highway Capacity Manual v/c and delay bar charts.

Transient Command: TOOLS - Sight Dist - Horizontal - Del Graphics

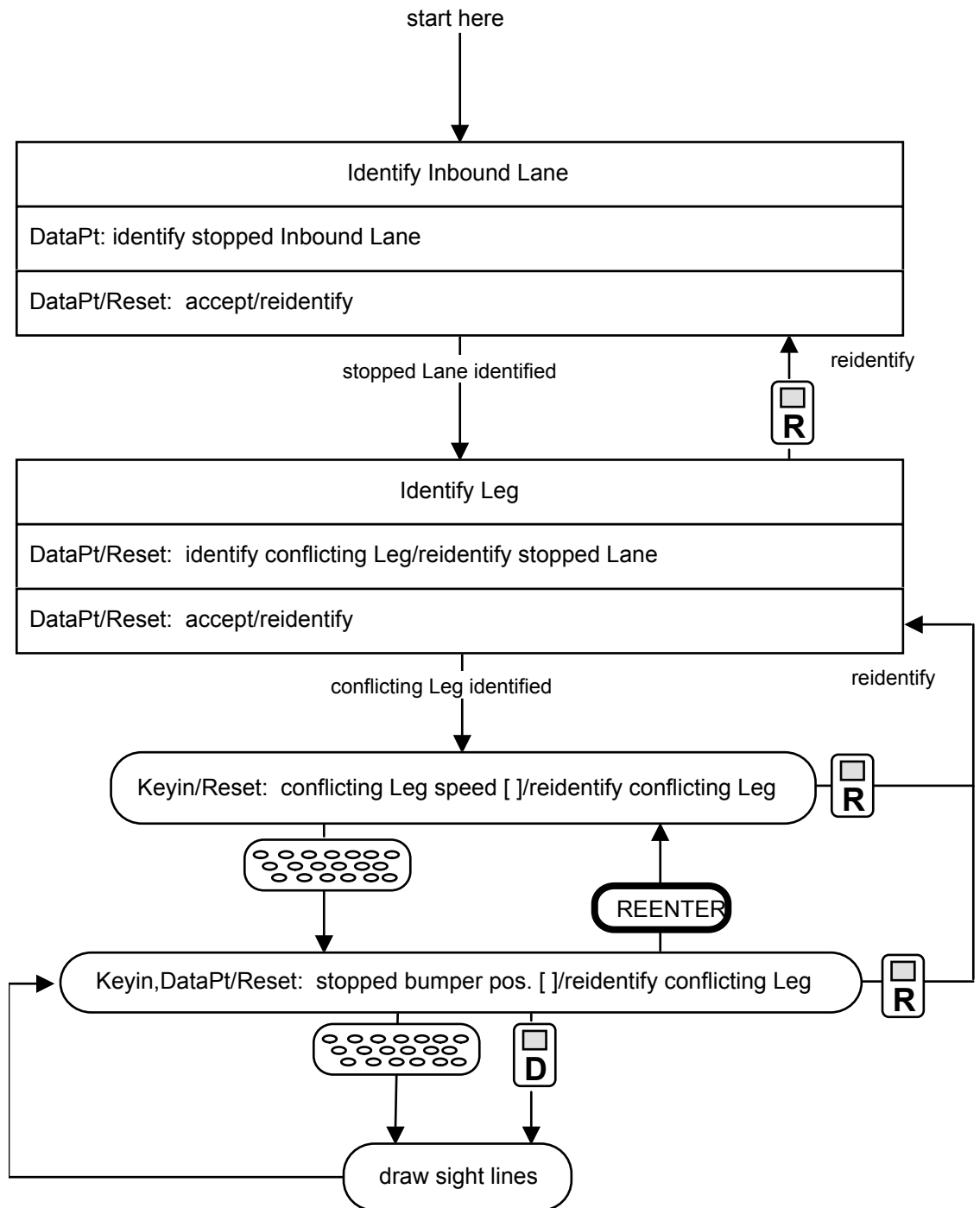
Delete all of the Horizontal Sight Distance graphics.

Primitive Command: TOOLS - Sight Dist - Horizontal - No Control



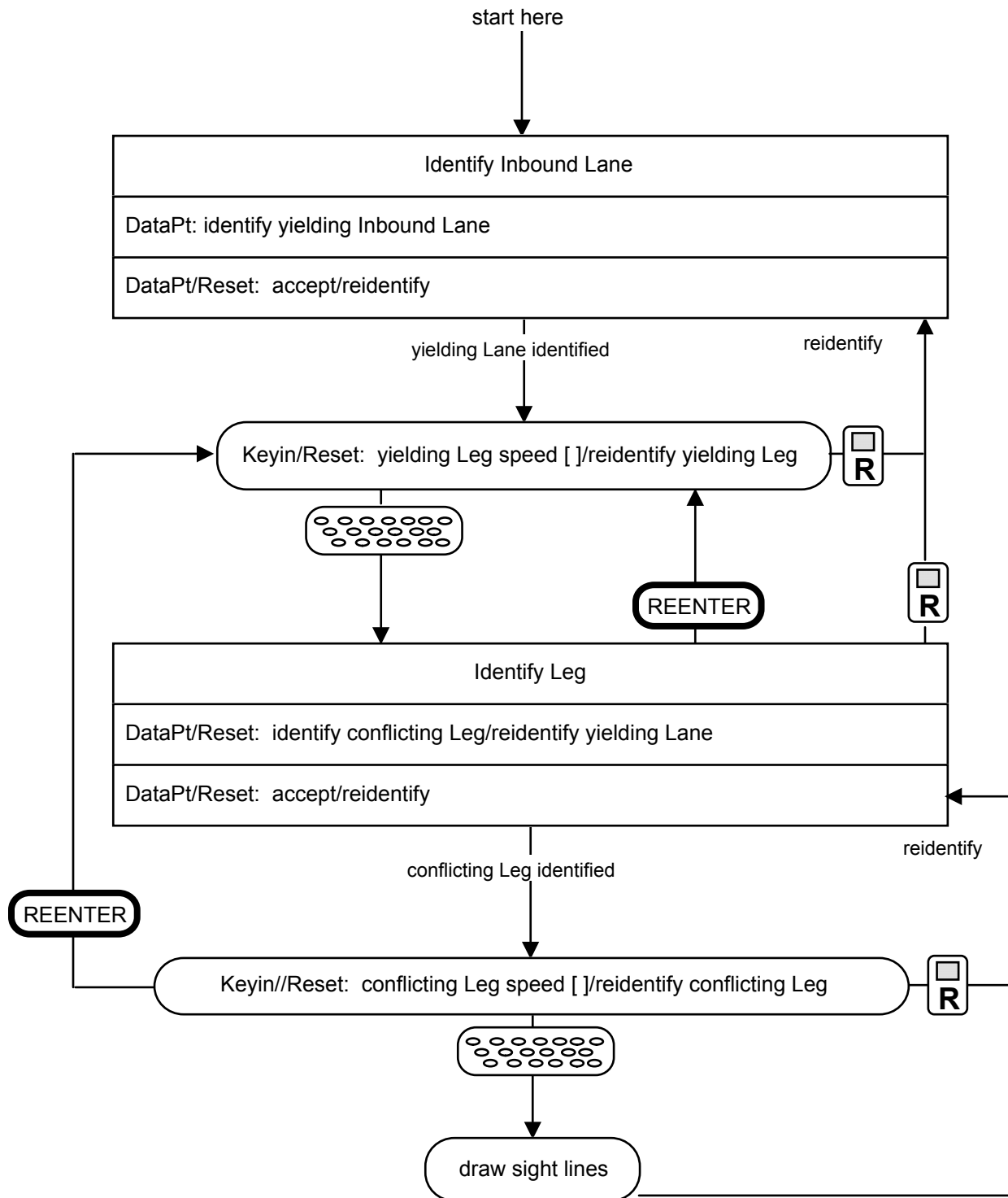
Draw horizontal sight line graphics for two approaches of an uncontrolled intersection.

Primitive Command: TOOLS - Sight Dist - Horizontal - Stopped



Draw horizontal sight line graphics for the intersection of a stop sign controlled approach and an uncontrolled approach.

Primitive Command: TOOLS - Sight Dist - Horizontal - Yield



Draw horizontal sight line graphics for the intersection of a yield sign controlled approach and an uncontrolled approach.

Transient Command: TOOLS - Sight Dist - Vertical - Del Graphics
Delete all of the Vertical Sight Distance graphics.

Primitive Command: TOOLS - Sight Dist - Vertical - No Control

DataPt: identify first Inbound Lane

keyin: not allowed

datapt: identify and accept first inbound lane

reset: not allowed

reenter: not allowed

Keyin/Reset: first Leg speed [default value]/reidentify first

keyin: double

datapt: not allowed

reset: backup to DataPt: identify first Inbound Lane

reenter: not allowed

DataPt/Reset: identify second Inbound Lane/reidentify first

keyin: not allowed

datapt: identify and accept second inbound lane

reset: backup to DataPt: identify first Inbound Lane

reenter: backup to Keyin/Reset: first Leg speed [default value]/reidentify

Keyin/Reset: second Leg speed [default value]/reidentify first

keyin: double

datapt: not allowed

reset: backup to Keyin/Reset: first Leg speed [default value]/reidentify first

reenter: not allowed

identify triangles for surface

accept triangles for surface

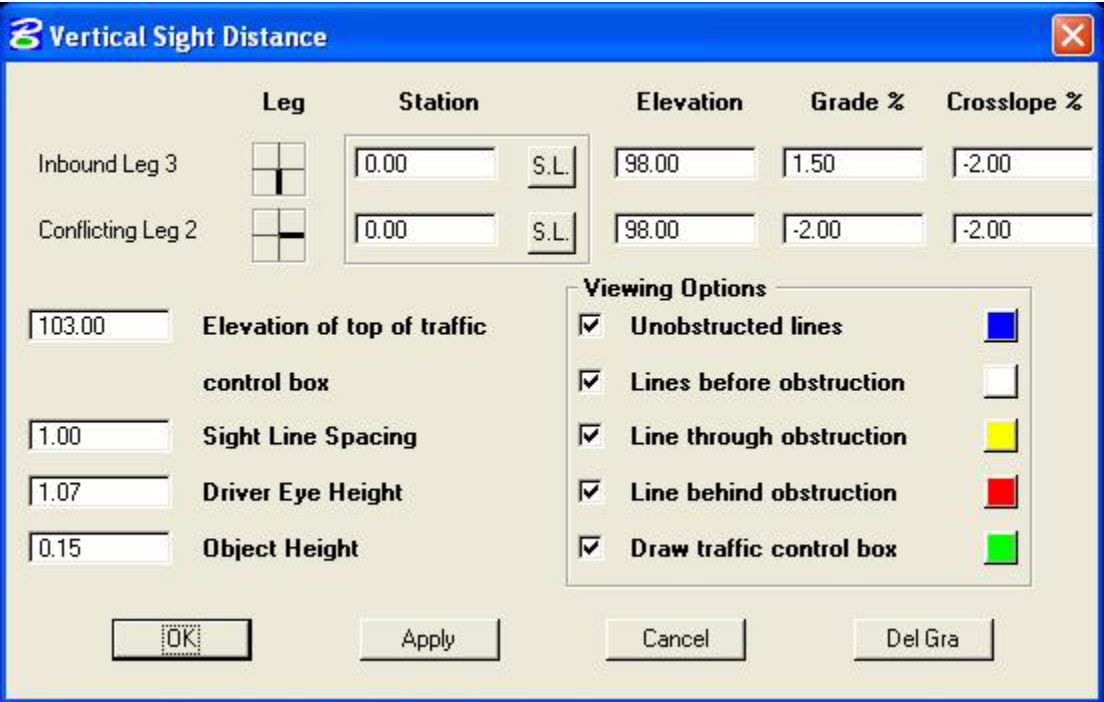
shape

keyin: not allowed



datapt: identify and accept a triangle in the dtm; open Vertical Sight Distance dialog box

reset: not allowed

reenter: not allowed



The dialog box is titled "Vertical Sight Distance" and contains a table for leg data, input fields for viewing parameters, and a viewing options section.

	Leg	Station	Elevation	Grade %	Crosslope %
Inbound Leg 3		0.00 S.L.	98.00	1.50	-2.00
Conflicting Leg 2		0.00 S.L.	98.00	-2.00	-2.00

Input fields for viewing parameters:

- Elevation of top of traffic control box: 103.00
- Sight Line Spacing: 1.00
- Driver Eye Height: 1.07
- Object Height: 0.15

Viewing Options:

- ☒ Unobstructed lines (blue icon)
- ☒ Lines before obstruction (white icon)
- ☒ Line through obstruction (yellow icon)
- ☒ Line behind obstruction (red icon)
- ☒ Draw traffic control box (green icon)

Buttons: OK, Apply, Cancel, Del Gra

Primitive Command: TOOLS - Sight Dist - Vertical - Stopped

DataPt: identify stopped Inbound Lane

keyin: not allowed

datap: identify and accept stopped inbound lane

reset: not allowed

reenter: not allowed

DataPt/Reset: identify conflicting Leg/reidentify

keyin: not allowed

datap: identify and accept conflicting leg

reset: backup to DataPt: identify stopped Inbound Lane

reenter: not allowed

Keyin/Reset: conflicting Leg speed [default value]/reidentify

keyin: double

datap: not allowed

reset: backup to DataPt/Reset: identify conflicting Leg/reidentify

reenter: not allowed

Keyin,DataPt/Reset: stopped bumper pos.[]/reidentify

keyin: double

datap: convert point to bumper position

reset: backup to DataPt/Reset: identify conflicting Leg/reidentify

reenter: backup to Keyin/Reset: conflicting Leg speed [default value]/reidentify

identify triangles for surface

accept triangles for surface

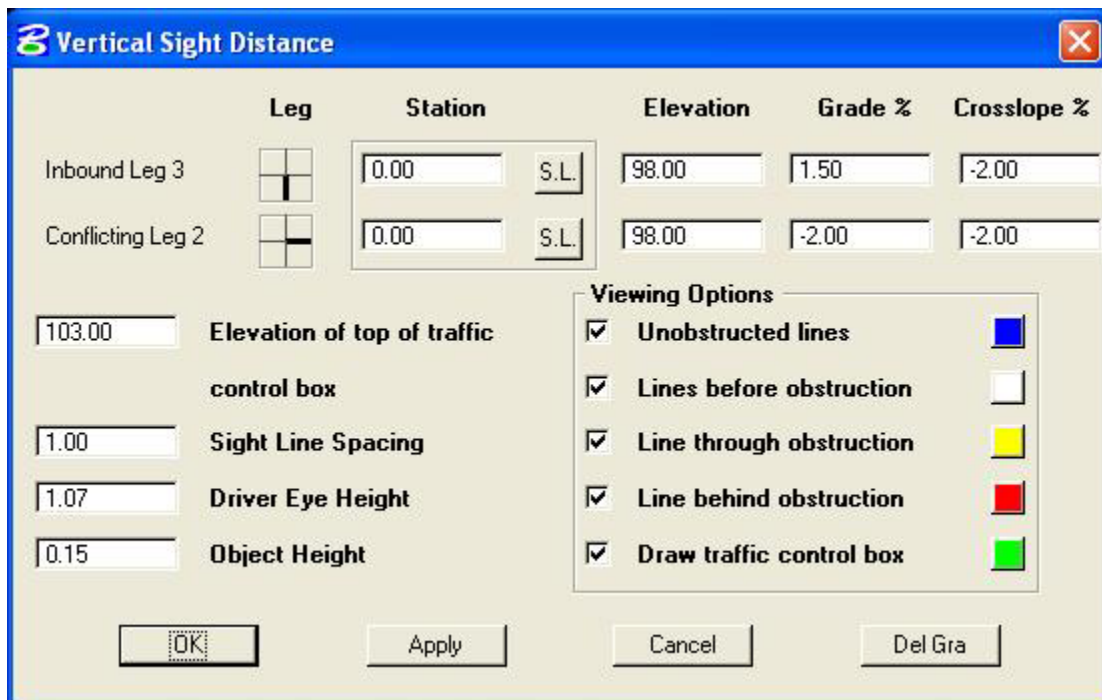
shape

keyin: not allowed



datap: identify and accept a triangle in the dtm; open Vertical Sight Distance dialog box

reset: not allowed

reenter: not allowed



The dialog box is titled "Vertical Sight Distance" and contains the following elements:

	Leg	Station	Elevation	Grade %	Crosslope %
Inbound Leg 3		0.00 S.L.	98.00	1.50	-2.00
Conflicting Leg 2		0.00 S.L.	98.00	-2.00	-2.00

Below the table, there are input fields for:

- Elevation of top of traffic control box: 103.00
- Sight Line Spacing: 1.00
- Driver Eye Height: 1.07
- Object Height: 0.15

On the right, the "Viewing Options" section includes:

- ☒ Unobstructed lines (blue square)
- ☒ Lines before obstruction (white square)
- ☒ Line through obstruction (yellow square)
- ☒ Line behind obstruction (red square)
- ☒ Draw traffic control box (green square)

At the bottom are buttons for OK, Apply, Cancel, and Del Gra.

Primitive Command: TOOLS - Sight Dist - Vertical - Yield

DataPt: identify yielding Inbound Lane

keyin: not allowed

datapt: identify and accept yielding inbound lane

reset: not allowed

reenter: not allowed

Keyin/Reset: yielding Leg speed [default value]/reidentify yielding

keyin: double

datapt: not allowed

reset: backup to DataPt: identify yielding Inbound Lane

reenter: not allowed

DataPt/Reset: identify conflicting Leg/reidentify yielding

keyin: not allowed

datapt: identify and accept conflicting leg

reset: backup to DataPt: identify yielding Inbound Lane

reenter: backup to Keyin/Reset: yielding Leg speed [default value]/reidentify

Keyin/Reset: conflicting Leg speed [default value]/reidentify yielding

keyin: double

datapt: not allowed

reset: backup to DataPt/Reset: identify conflicting Leg/reidentify yielding

reenter: backup to Keyin/Reset: yielding Leg speed [default value]/reidentify yielding

identify triangles for surface

accept triangles for surface

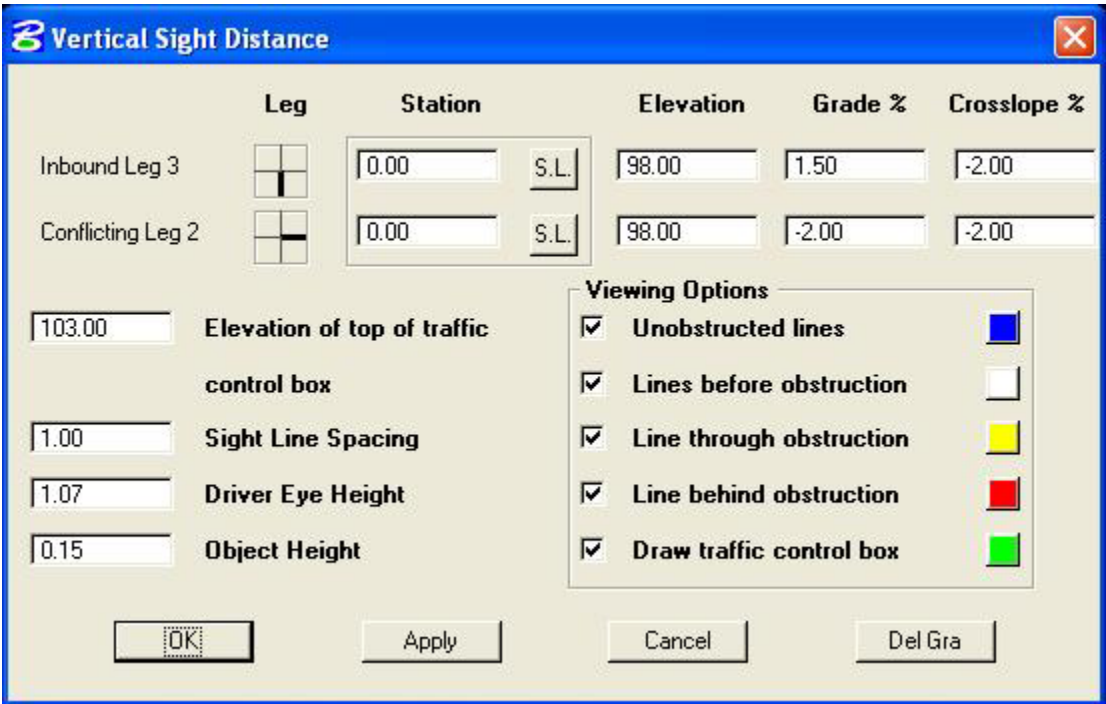
shape

keyin: not allowed

datapt: identify and accept a triangle in the dtm; open Vertical Sight Distance dialog box

reset: not allowed

reenter: not allowed



The dialog box is titled "Vertical Sight Distance" and contains several input fields and checkboxes. It is organized into sections for leg data, viewing options, and control parameters.

Leg	Station	Elevation	Grade %	Crosslope %
Inbound Leg 3	0.00 S.L.	98.00	1.50	-2.00
Conflicting Leg 2	0.00 S.L.	98.00	-2.00	-2.00

Viewing Options

- ☒ Unobstructed lines
- ☒ Lines before obstruction
- ☒ Line through obstruction
- ☒ Line behind obstruction
- ☒ Draw traffic control box

Control Parameters

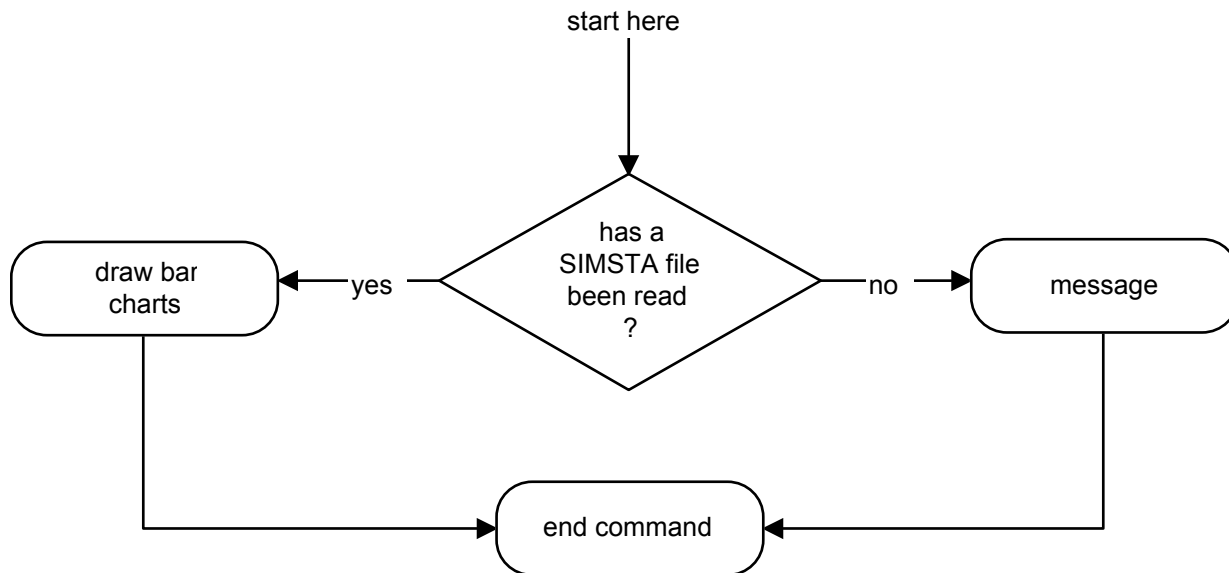
- Elevation of top of traffic control box: 103.00
- Sight Line Spacing: 1.00
- Driver Eye Height: 1.07
- Object Height: 0.15

Buttons: OK, Apply, Cancel, Del Gra

Primitive Command: TOOLS - TEXAS Model - Animation
Not programmed yet.

Transient Command: TOOLS - TEXAS Model - Del Graphics
Delete all of the TEXAS Model statistics bar charts.

Primitive Command: TOOLS - TEXAS Model - Graph



Use this diagram for the 14 **TOOLS - TEXAS Model - Graph** commands that follow. The user must have previously executed the **Primitive Command: TOOLS - TEXAS Model - Load SIMSTA** before displaying statistical data from the TEXAS Model. It is the responsibility of the user to be sure that the statistics file matches the selected Alternative.

Primitive Command: TOOLS - TEXAS Model - Graph ADMPH

Draw bar charts showing the Average Delay Below XX MPH statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph AQD

Draw bar charts showing the Average Queue Delay statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph ASD

Draw bar charts showing the Average Stopped Delay statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph ATD

Draw bar charts showing the Average Total Delay statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph DMPH

Draw bar charts showing the Delay Below XX MPH statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph OADMPH

Draw bar charts showing the Overall Average Delay Below XX MPH statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph OAQD

Draw bar charts showing the Overall Average Queue Delay statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph OASD

Draw bar charts showing the Overall Average Stopped Delay statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph OATD

Draw bar charts showing the Overall Average Total Delay statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph Probs

Draw bar chart showing the 95 percent confidence interval statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph QD

Draw bar charts showing the Queue Delay statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph Queues

Draw bar charts showing the Maximum and Average Queue Length statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph SD

Draw bar charts showing the Stopped Delay statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph TD

Draw bar charts showing the Total Delay statistics from a TEXAS Model analysis.

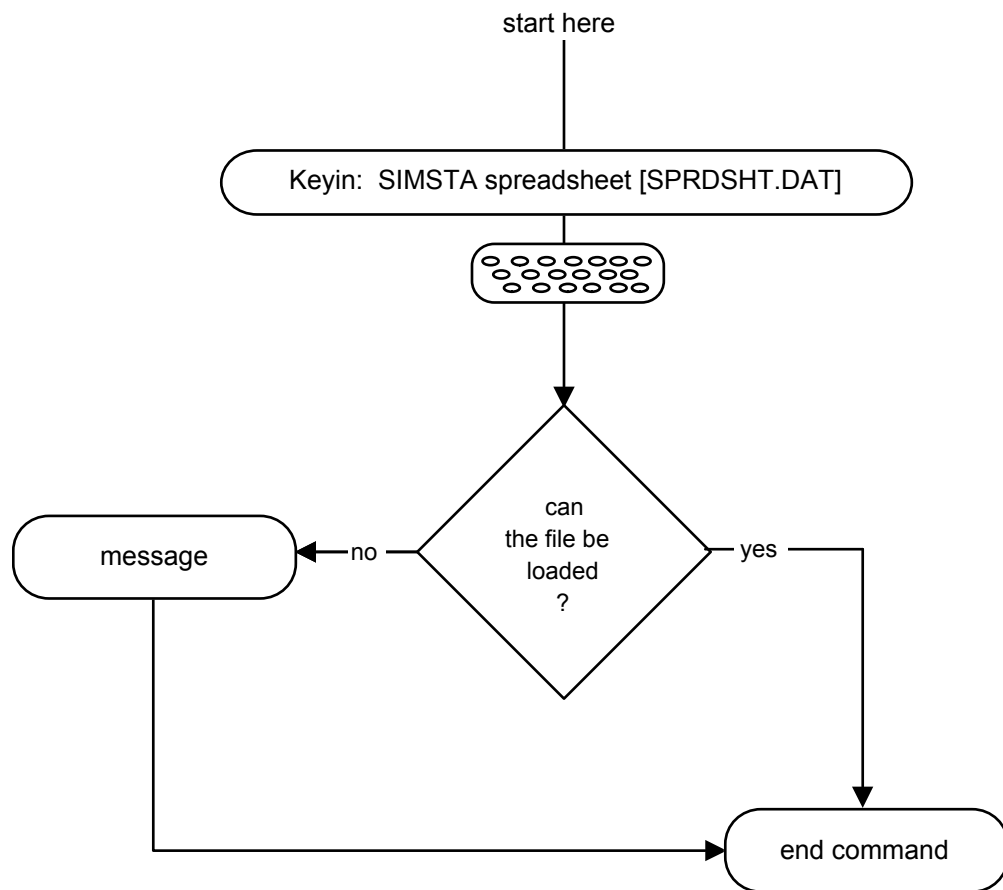
Primitive Command: TOOLS - TEXAS Model - Graph Turn %

Draw bar charts showing the Percent of Vehicles Making Turning Movements statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Graph Volume

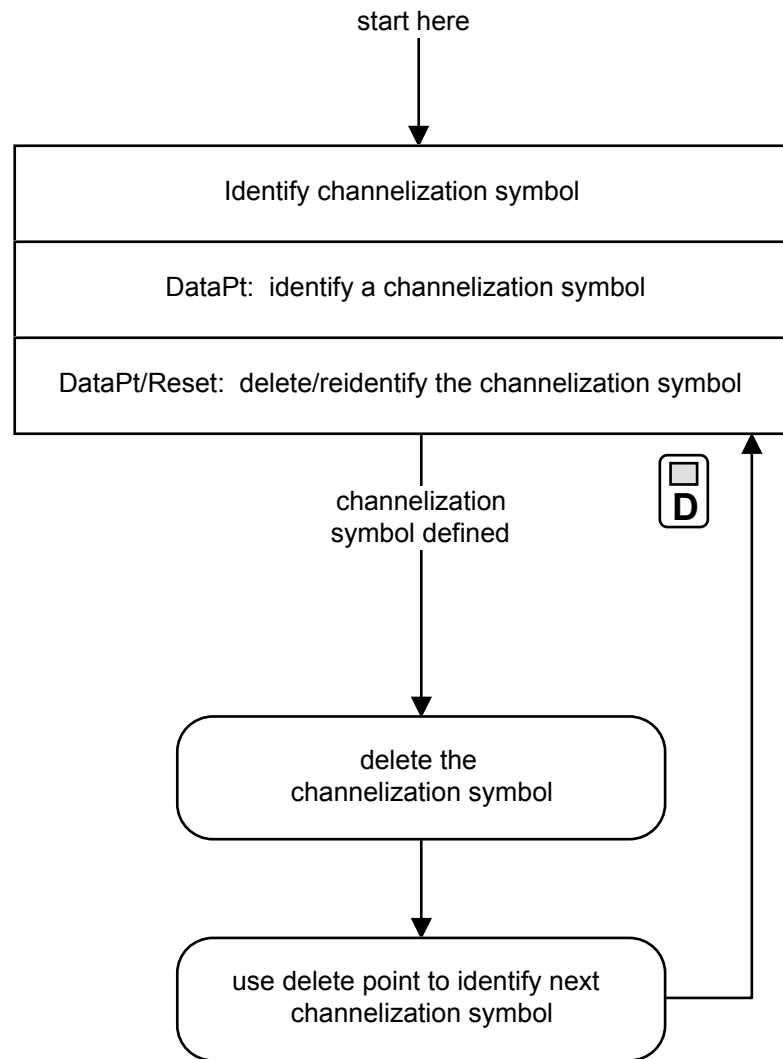
Draw bar charts showing the Volume Processed statistics from a TEXAS Model analysis.

Primitive Command: TOOLS - TEXAS Model - Load SIMSTA



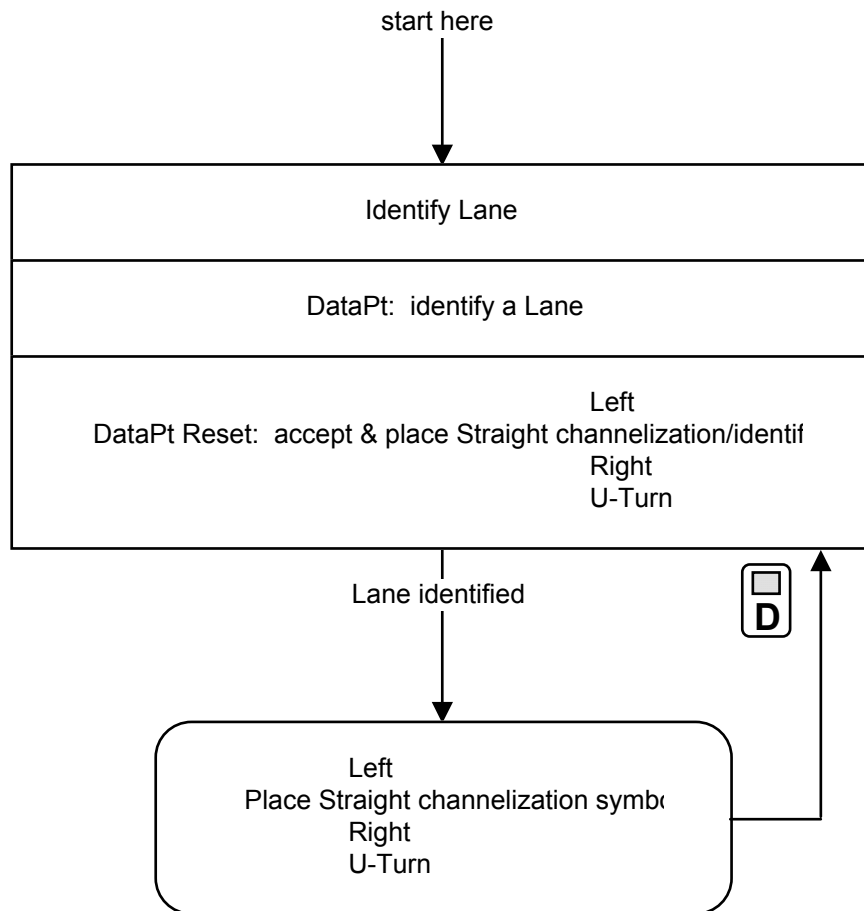
Load statistical data from file created by the TEXAS Model. This load must be done before displaying statistical data from the TEXAS Model. It is the responsibility of the user to be sure that the statistics file matches the selected Alternative.

Primitive Command: TOOLS - Traffic - Channelize - DELETE



Delete an existing channelization symbol.

Primitive Command: TOOLS - Traffic - Channelize



Use this diagram for the 4 **TOOLS - Traffic - Channelize** commands that follow.

Primitive Command: TOOLS - Traffic - Channelize - Left

Add a left turn channelization symbol to a Lane.

Primitive Command: TOOLS - Traffic - Channelize - Right

Add a right turn channelization symbol to a Lane.

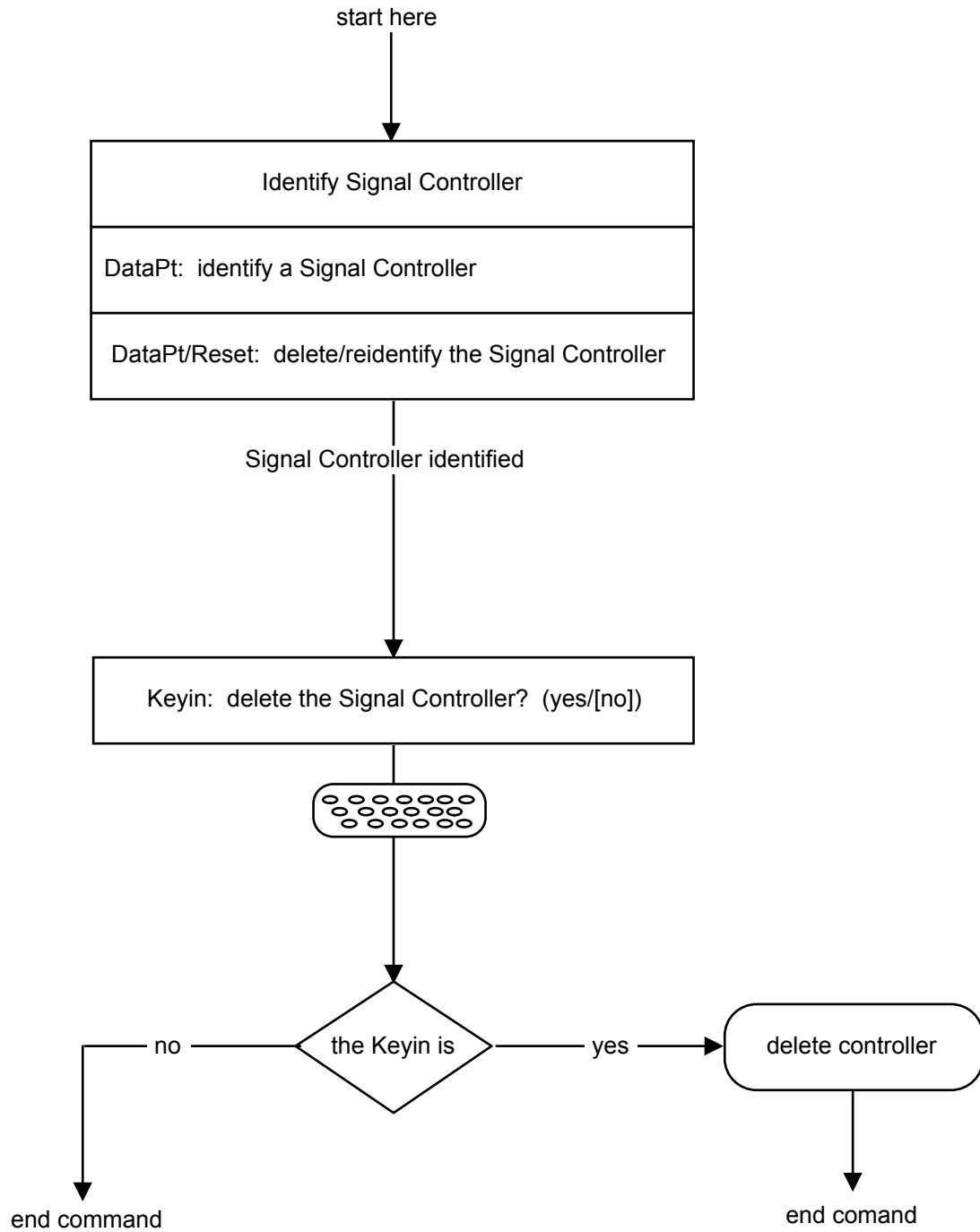
Primitive Command: TOOLS - Traffic - Channelize - Straight

Add a straight through channelization symbol to a Lane.

Primitive Command: TOOLS - Traffic - Channelize - U-Turn

Add a u-turn channelization symbol to a Lane.

Primitive Command: TOOLS - Traffic - Controller - DELETE



Delete an existing traffic signal controller.

Primitive Command: TOOLS - Traffic - Controller - MOVE

identify Traffic Control/end command

keyin: not allowed

datapt: select object

reset: end command

reenter: not allowed

DataPt: NEW Traffic Control placement point

keyin: precision keyin allowed

datapt: new placement point

reset: use current placement point for new placement point

reenter: backup to identify Traffic Control/end command

DataPt: New Traffic Control angle, 1st point

keyin: precision keyin allowed

datapt: 1st point for angle

reset: use current angle for new angle

reenter: backup to DataPt: NEW Traffic Control placement point

if not reset on last command then DataPt: New Traffic Control angle, 2nd point

keyin: precision keyin allowed

datapt: 2nd point for angle

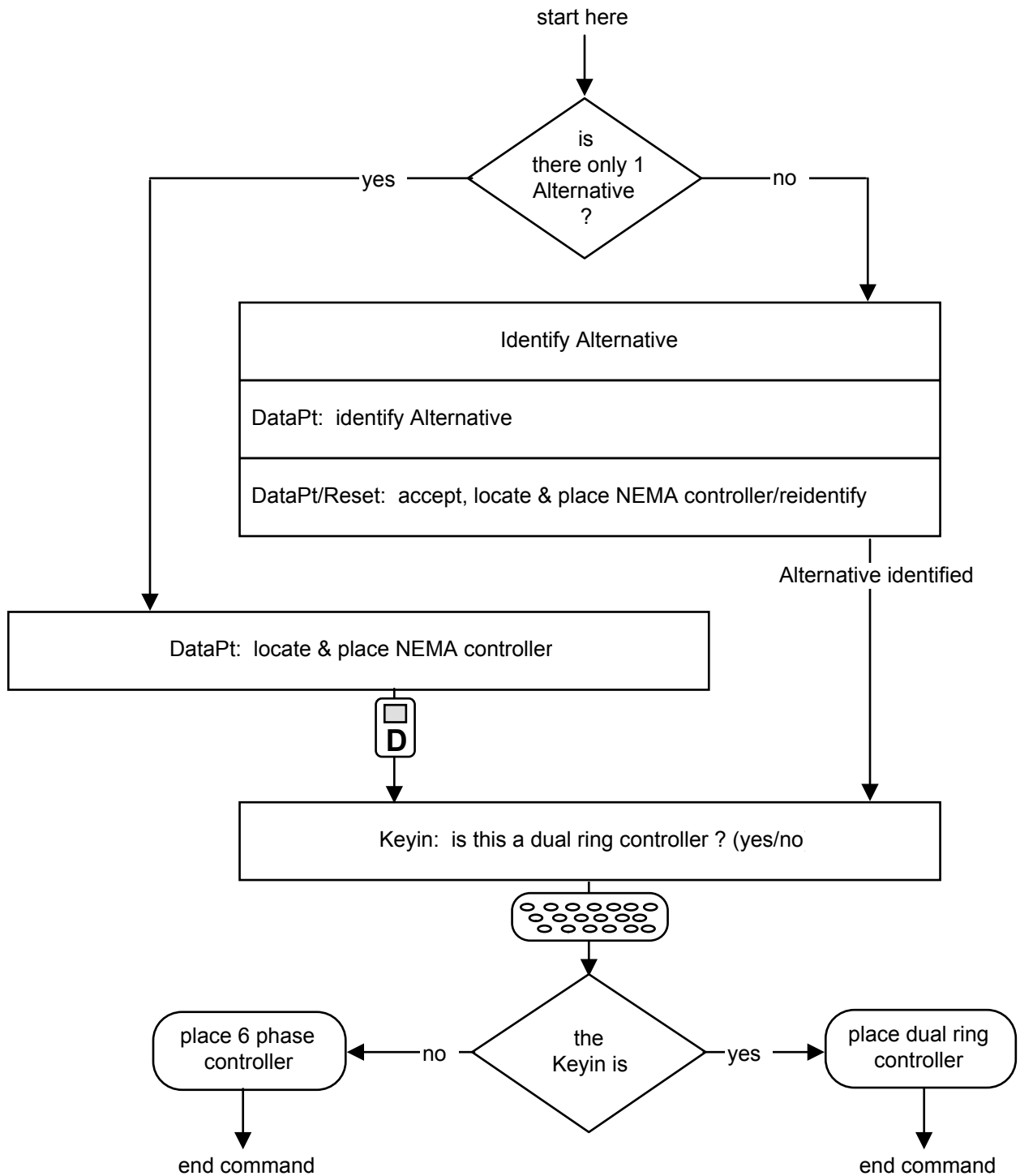
reset: use current angle for new angle

reenter: backup to DataPt: NEW Traffic Control placement point

selected traffic control symbol is moved and rotated.

This is the same as **Primitive Command: MOVE - TRAF CONTRL**, **Primitive Command: TOOLS - Traffic - Sign - MOVE**, and **Primitive Command: TOOLS - Traffic - Signal Face - MOVE**.

Primitive Command: TOOLS - Traffic - Controller - NEMA



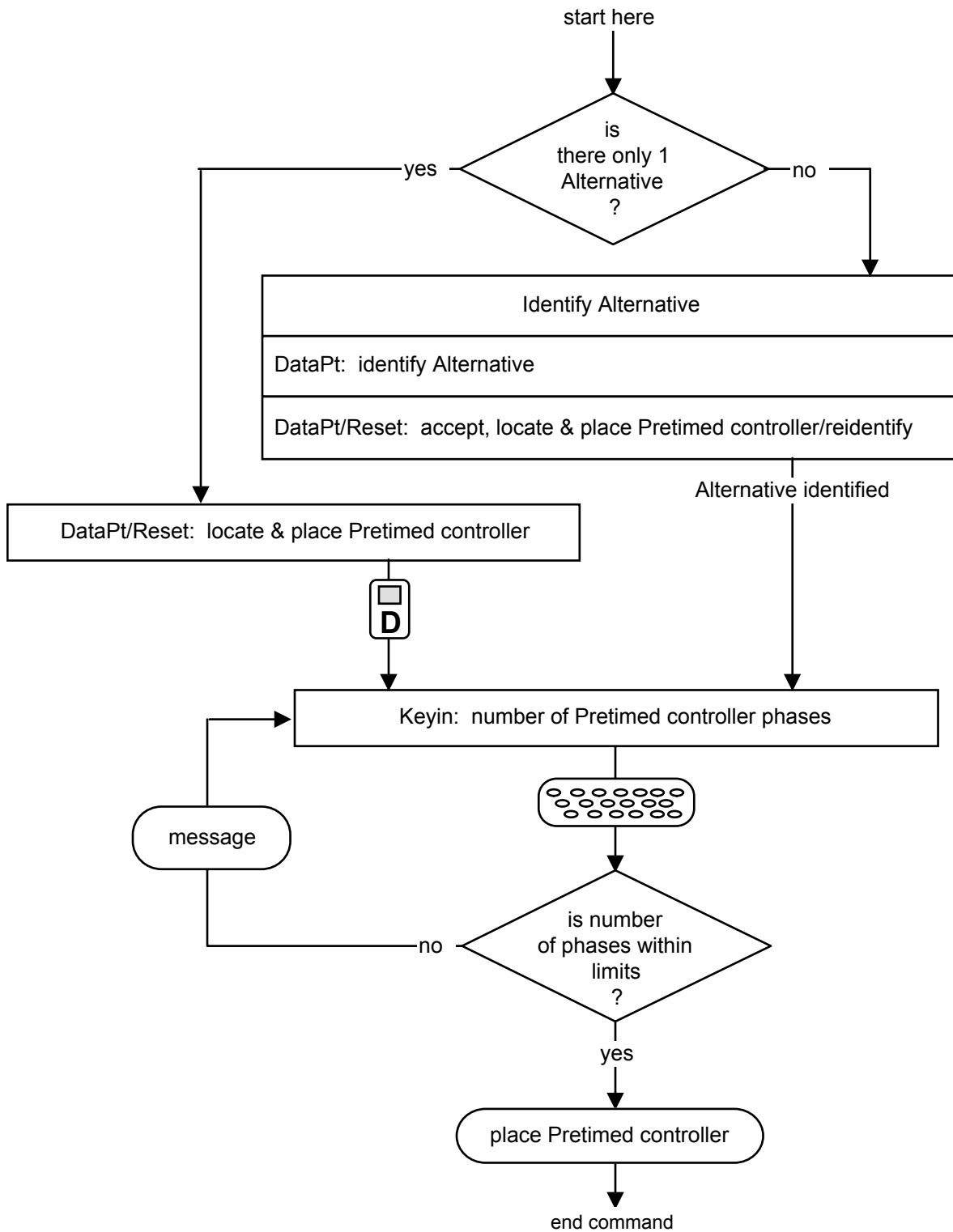
Add a standard NEMA traffic signal controller to an Alternative. The user is asked if the controller is to be a dual ring controller. If the response is "NO", a 6 phase single ring controller is added.

Specify the traffic phases to be associated with each controller phase. The user is prompted for a controller phase. The user is then prompted to identify signal faces and/or channelization symbols to be added or removed from the list for the controller phase. A circular green will permit all movements, except an exclusive left that does not move in another phase, to move. A protected left will permit



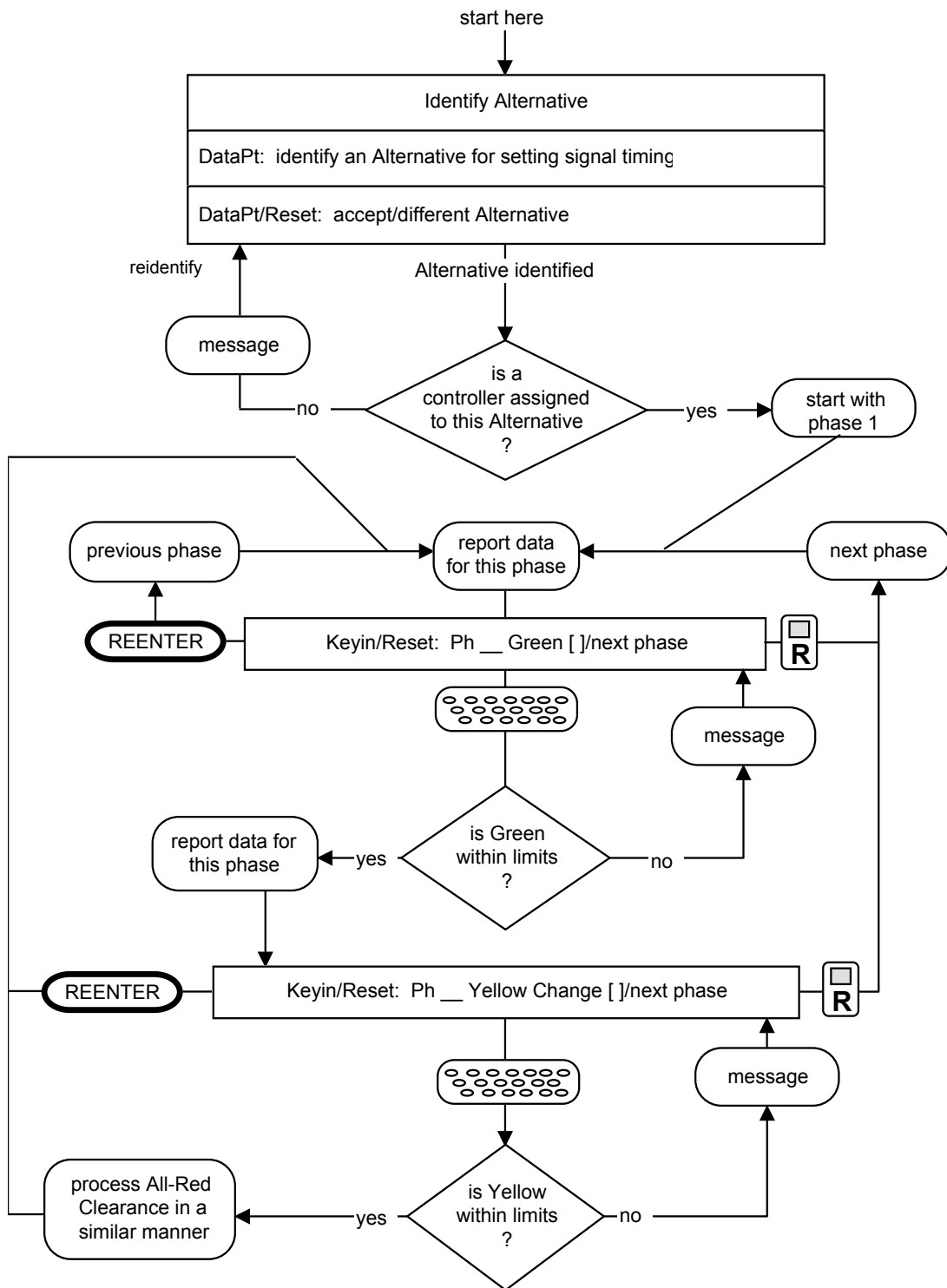
exclusive lefts to move. Identifying selected channelization symbols in addition to signal faces will modify the above. Items on the list are shown in green.

Primitive Command: TOOLS - Traffic - Controller - Pretimed



Add a pretimed traffic signal controller to an Alternative. The user is prompted for the number of controller phases. Must be 2 through 8.

Primitive Command: TOOLS - Traffic - Controller - TIMING



Specify the phase timing for a pretimed controller. The user is prompted (one interval at a time) for the green interval, yellow change interval and all-red clearance interval for a phase. Use Reset to go the next phase. This command is not yet programmed for a NEMA controller.

Primitive Command: TOOLS - Traffic - Inventory - Report

identify alternative

accept alternative/reidentify

alternative not found

keyin: not allowed

datapt: select alternative

reset: prompt again

reenter: prompt again

IGIDS Inventory Report; default = inventory.txt

OK: create file

cancel: end command

Creates an inventory report of the traffic control including striping suitable for printing.

Primitive Command: TOOLS - Traffic - Inventory - Spreadsheet

identify alternative

accept alternative/reidentify

alternative not found

keyin: not allowed

datapt: select alternative

reset: prompt again

reenter: prompt again

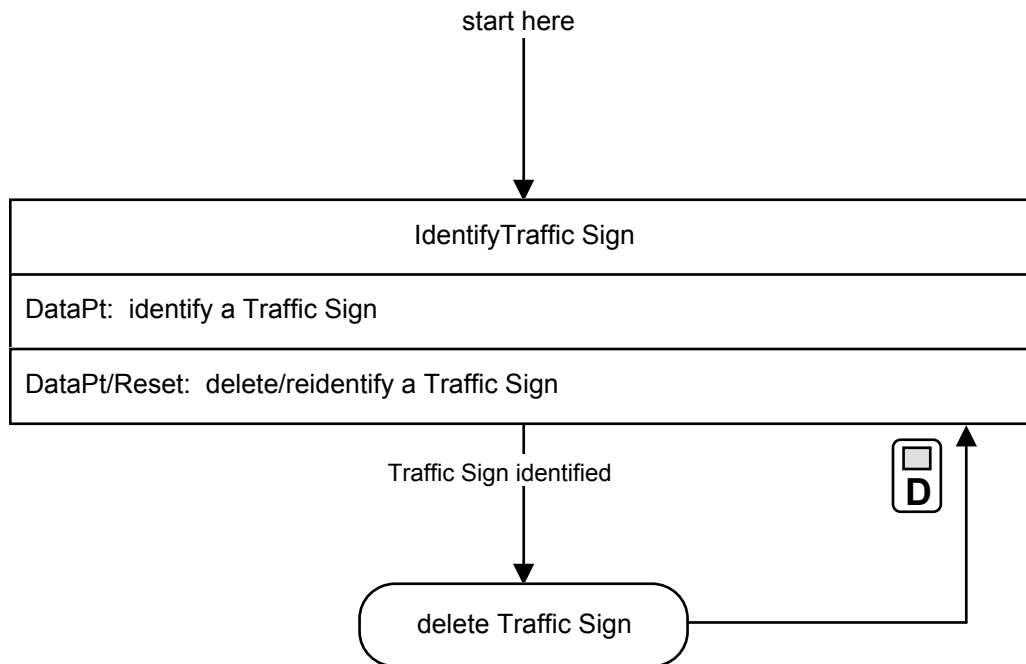
IGIDS Inventory Spreadsheet; default = inventory.dat

OK: create file

cancel: end command

Creates an inventory report of the traffic control including striping in comma-separated file format for importing into a database or spreadsheet program.

Primitive Command: TOOLS - Traffic - Sign - DELETE



Delete an existing stop or yield sign

Primitive Command: TOOLS - Traffic - Sign - MOVE

identify Traffic Control/end command

keyin: not allowed

datapt: select object

reset: end command

reenter: not allowed

DataPt: NEW Traffic Control placement point

keyin: precision keyin allowed

datapt: new placement point

reset: use current placement point for new placement point

reenter: backup to identify Traffic Control/end command

DataPt: New Traffic Control angle, 1st point

keyin: precision keyin allowed

datapt: 1st point for angle

reset: use current angle for new angle

reenter: backup to DataPt: NEW Traffic Control placement point

if not reset on last command then DataPt: New Traffic Control angle, 2nd point

keyin: precision keyin allowed

datapt: 2nd point for angle

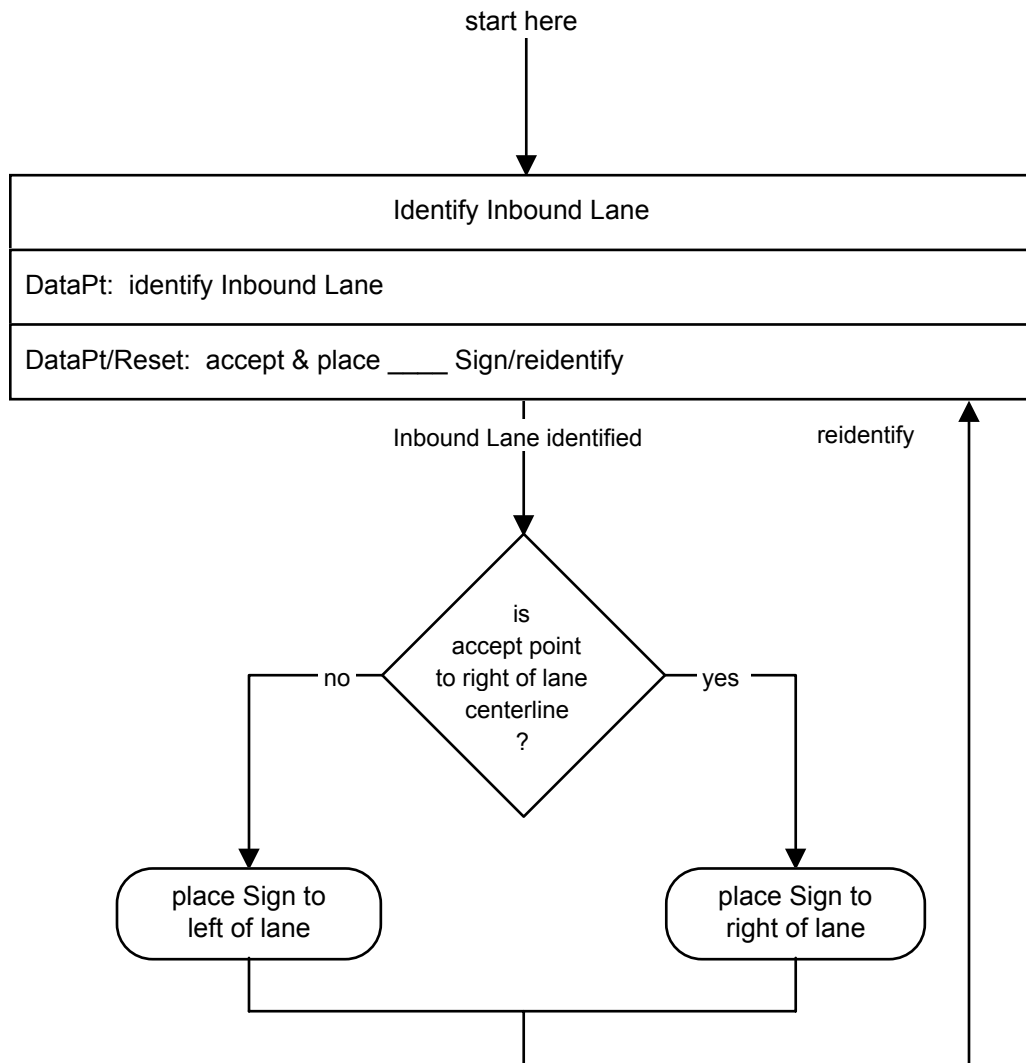
reset: use current angle for new angle

reenter: backup to DataPt: NEW Traffic Control placement point

selected traffic sign is moved and rotated.

This is the same as **Primitive Command: MOVE - TRAF CONTRL**, **Primitive Command: TOOLS - Traffic - Controller - MOVE**, and **Primitive Command: TOOLS - Traffic - Signal Face - MOVE**.

Primitive Command: TOOLS - Traffic - Sign



Use this diagram for the 2 **TOOLS - Traffic - Sign** commands that follow.

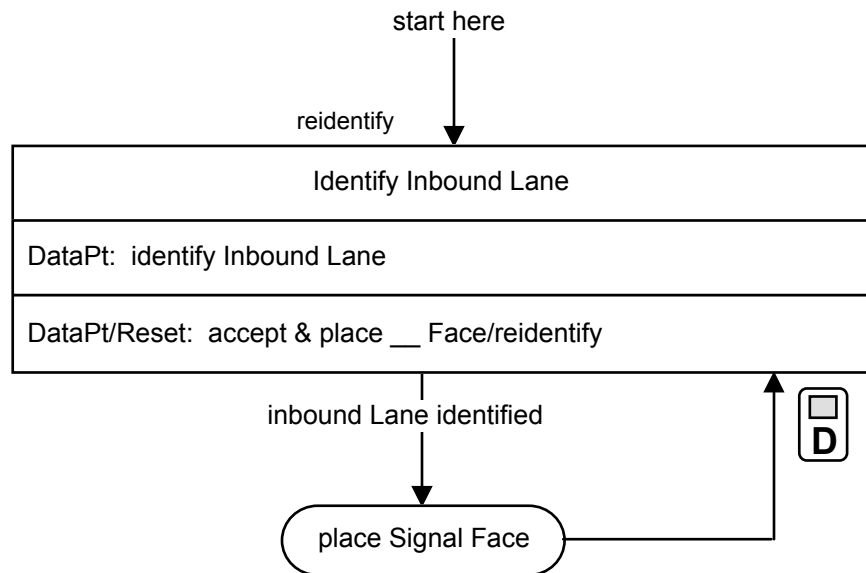
Primitive Command: TOOLS - Traffic - Sign - Stop

Add a stop sign to an inbound Lane. The location of the acceptance data point with respect to the center of the lane will determine if the sign is located to the left or right of the lane.

Primitive Command: TOOLS - Traffic - Sign - Yield

Add a yield sign to an inbound Lane. The location of the acceptance data point with respect to the center of the lane will determine if the sign is located to the left or right of the lane.

Primitive Command: TOOLS - Traffic - Signal Face



Use this diagram for the 3 **TOOLS - Traffic - Signal Face** commands that follow.

Primitive Command: TOOLS - Traffic - Signal Face - 3 Lens

Add a 3 lens signal face to a lane. This face presents a circular green to the appropriate movements on the Leg.

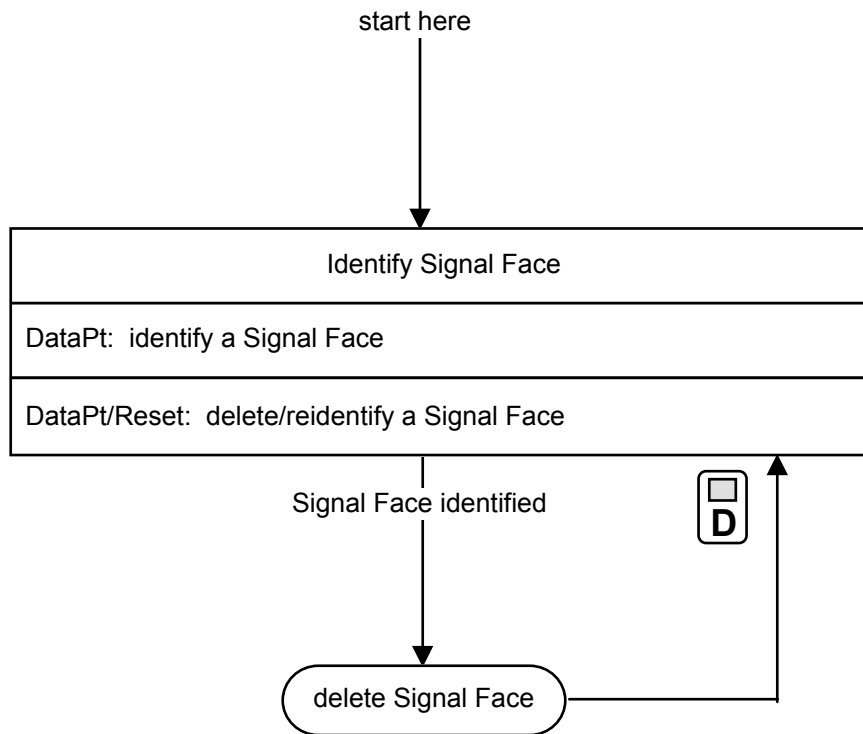
Primitive Command: TOOLS - Traffic - Signal Face - 3 Lens PL

Add a 3 lens signal face to a lane. This face presents a left green arrow green to the appropriate movements on the Leg.

Primitive Command: TOOLS - Traffic - Signal Face - 4 Lens

Add a 4 lens signal face to a lane. Don't use this command. Place a 3 lens signal face, instead.

Primitive Command: TOOLS - Traffic - Signal Face - DELETE



Delete an existing signal face.

Primitive Command: TOOLS - Traffic - Signal Face - MOVE

identify Traffic Control/end command

keyin: not allowed

datapt: select object

reset: end command

reenter: not allowed

DataPt: NEW Traffic Control placement point

keyin: precision keyin allowed

datapt: new placement point

reset: use current placement point for new placement point

reenter: backup to identify Traffic Control/end command

DataPt: New Traffic Control angle, 1st point

keyin: precision keyin allowed

datapt: 1st point for angle

reset: use current angle for new angle

reenter: backup to DataPt: NEW Traffic Control placement point

if not reset on last command then DataPt: New Traffic Control angle, 2nd point

keyin: precision keyin allowed

datapt: 2nd point for angle

reset: use current angle for new angle

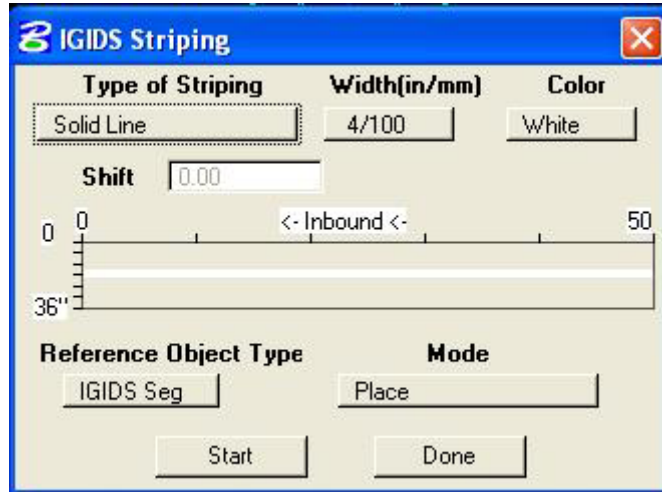
reenter: backup to DataPt: NEW Traffic Control placement point

selected traffic signal face is moved and rotated.

This is the same as **Primitive Command: MOVE - TRAF CONTRL**, **Primitive Command: TOOLS - Traffic - Controller - MOVE**, and **Primitive Command: TOOLS - Traffic - Sign - MOVE**.

Primitive Command: TOOLS - Traffic - STRIPING

Opens the IGIDS Striping dialog box and allows the user to place striping, modify the attributes of striping, and delete striping.



Primitive Command: TOOLS - Traffic - Volume

A different dialog box is presented based upon the number of legs (3, 4, 5, or 6) for the Alternative. All Traffic Volume dialog boxes have the same general layout and functionality. The dialog boxes for the 4 leg Alternative are somewhat different from the others. The input mode of each dialog box may be changed between **Percent+VOL** (Percentages of Traffic Volume) and **TMC Volume** (Traffic Volumes: Vehicles per Hour) when the data will allow the transfer using the option button labeled "INPUT MODE" in the top center of the dialog box. In all dialog boxes, each inbound leg has a row of input data boxes. The leg number and a small diagram of the Alternative with the leg highlighted are to the left of the input data boxes. The input data boxes may be traversed from left to right by using the tab key on the keyboard. A tab will move the cursor from the rightmost input data box on one row to the leftmost input data box on the next row. Standard MicroStation dialog box editing functions may be used to traverse and edit the data. The push button labeled "Cancel" may be pressed at anytime and the input will be discarded and the dialog box closed. When all input data has been entered and is correct, a push button labeled "OK" will appear in the lower left of the dialog box. Pressing the push button labeled "OK" will save the input data to the Alternative and close the dialog box. When the input mode is **Percent+VOL**, there is a column labeled "Total Percent" which is the sum of the percentages on the row and there is a column labeled "Total Volume" which is an input data box. When the input mode is **TMC Volume**, there is a column labeled "Total Volume" which is the sum of the volumes on the row.

In the 3, 5, and 6 leg dialog boxes, each outbound leg has a column of input data boxes. The leg number and a small diagram of the Alternative with the leg highlighted are to the top of the input data boxes. The inbound legs are sorted clockwise with the north leg at the top. The outbound legs are sorted clockwise with the north leg at the left. Each input data box in the matrix applies to traffic traveling from the inbound leg to the outbound leg. Additionally, there is an option button with the values "U" (u-turn), "L" (left), "S" (straight), or "R" (right) to the right of each input data box for the leg data. This option button indicates the designation of the traffic movement when a vehicle travels from the inbound leg to the outbound leg. Designations considered impossible are disabled. This designation may be changed by the user causing each designation on a row to be re-evaluated.

In the 4 leg dialog box, the rows are labeled "EB" (east bound), "WB" (west bound), "NB" (north bound), and "SB" (south bound) while the columns are labeled "U-Turn", "Left Turn", "Straight", and "Right".

These dialog boxes are used for the 8 **TOOLS - Traffic - Volume** commands that follow.

Primitive Command: TOOLS - Traffic - Volume - Percent+VOL (3 legs)

Primitive Command: TOOLS - Traffic - Volume - Percent+VOL (4 legs)

Primitive Command: TOOLS - Traffic - Volume - Percent+VOL (5 legs)

Primitive Command: TOOLS - Traffic - Volume - Percent+VOL (6 legs)

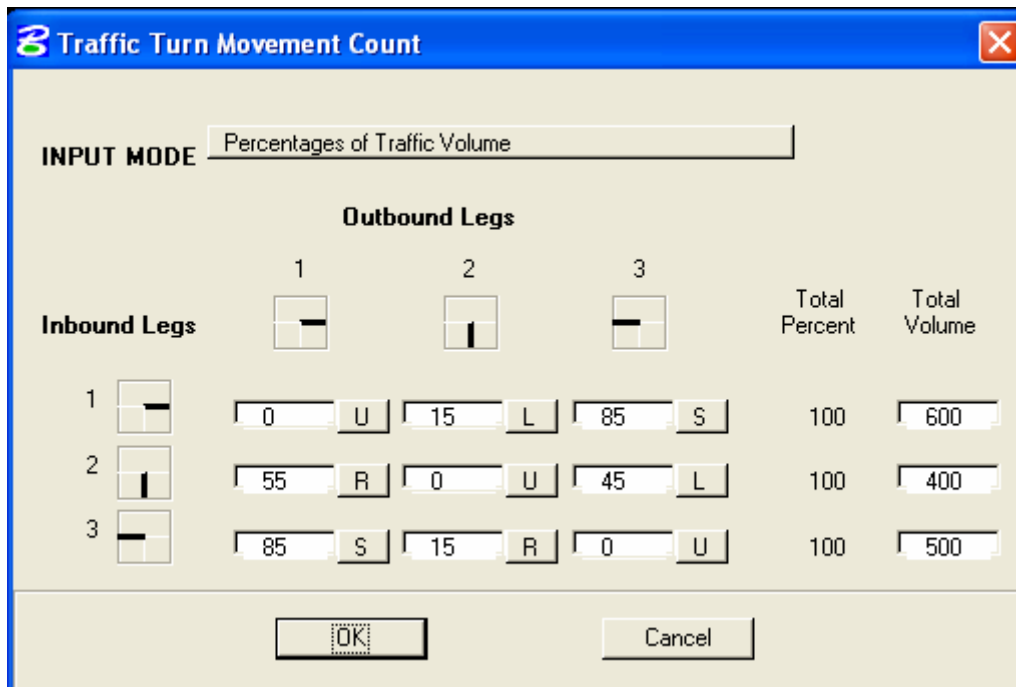
Primitive Command: TOOLS - Traffic - Volume - TMC Volume (3 legs)

Primitive Command: TOOLS - Traffic - Volume - TMC Volume (4 legs)







Primitive Command: TOOLS - Traffic - Volume - TMC Volume (5 legs)

Primitive Command: TOOLS - Traffic - Volume - TMC Volume (6 legs)

Primitive Command: TOOLS - Traffic - Volume - Percent+VOL (3 legs)

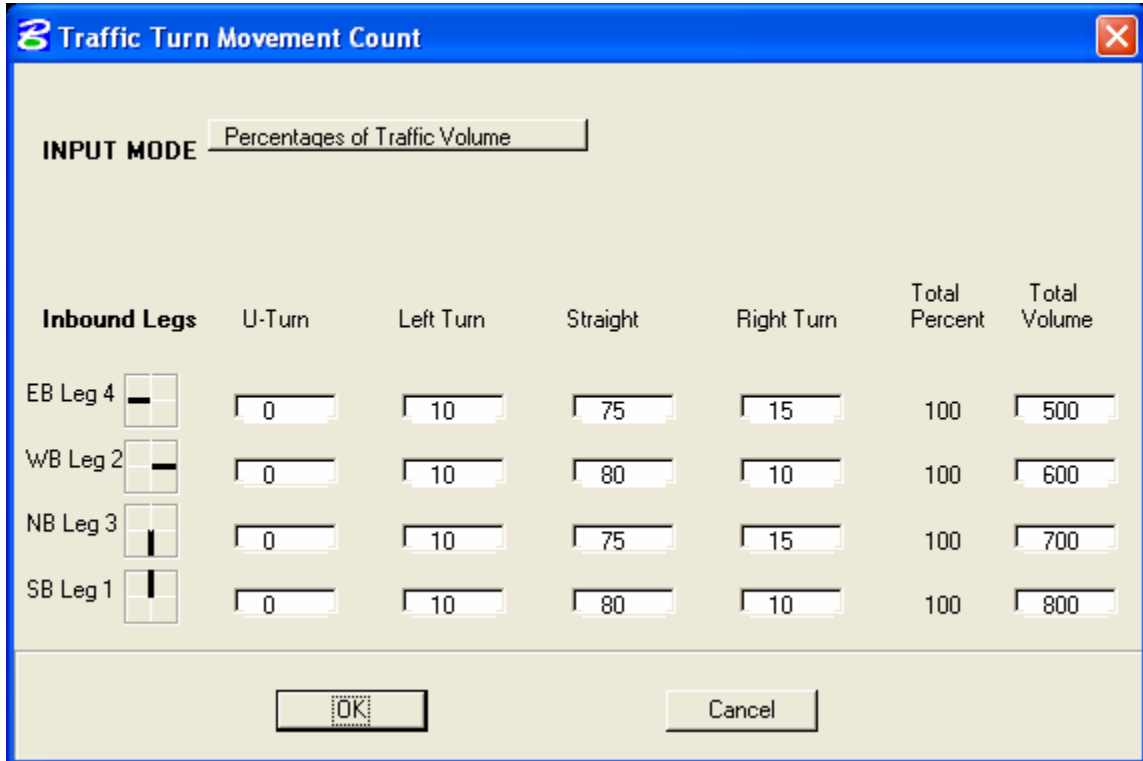


The dialog box is titled "Traffic Turn Movement Count" and has a close button (X) in the top right corner. It features an "INPUT MODE" dropdown menu set to "Percentages of Traffic Volume". Below this, there are three columns for "Outbound Legs" labeled 1, 2, and 3. To the left of these columns are three rows for "Inbound Legs" labeled 1, 2, and 3. Each cell in the grid contains a traffic movement diagram (a square with a horizontal or vertical line and an arrow) and a text input field for percentages. To the right of the grid are two columns: "Total Percent" and "Total Volume". At the bottom of the dialog are "OK" and "Cancel" buttons.

		Outbound Legs				
		1	2	3		
Inbound Legs					Total Percent	Total Volume
1		0 U	15 L	85 S	100	600
2		55 R	0 U	45 L	100	400
3		85 S	15 R	0 U	100	500

Specify percentages of traffic volumes plus the traffic volumes for a 3 leg Alternative. See the description of dialog box usage under **Primitive Command: TOOLS - Traffic - Volume**.

Primitive Command: TOOLS - Traffic - Volume - Percent+VOL (4 legs)



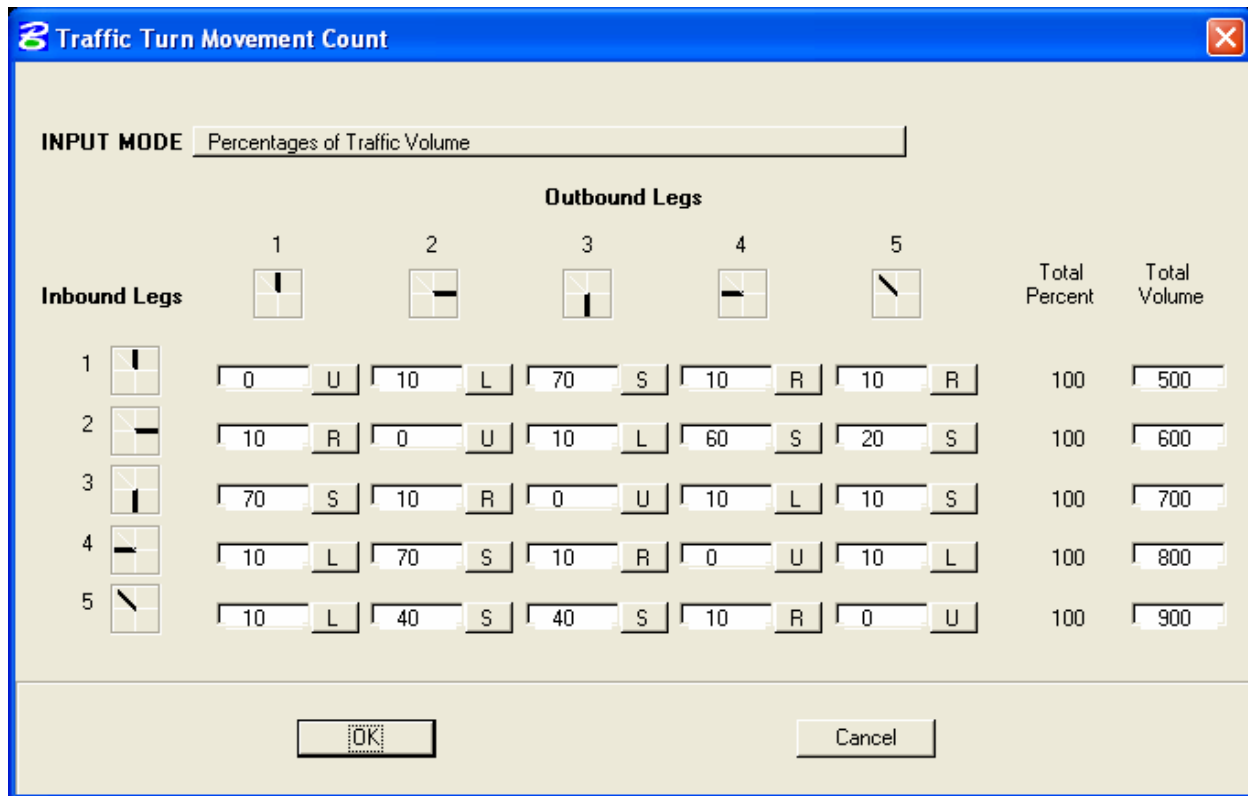
The dialog box is titled "Traffic Turn Movement Count" and features a close button in the top right corner. Below the title bar, the "INPUT MODE" is set to "Percentages of Traffic Volume". The main area contains a table for inputting traffic data for four inbound legs. Each leg has a diagram of a four-quadrant intersection with a specific movement highlighted: EB Leg 4 (Eastbound Left Turn), WB Leg 2 (Westbound Right Turn), NB Leg 3 (Northbound Left Turn), and SB Leg 1 (Southbound Left Turn). The table columns are "Inbound Legs", "U-Turn", "Left Turn", "Straight", "Right Turn", "Total Percent", and "Total Volume".

Inbound Legs	U-Turn	Left Turn	Straight	Right Turn	Total Percent	Total Volume
EB Leg 4	0	10	75	15	100	500
WB Leg 2	0	10	80	10	100	600
NB Leg 3	0	10	75	15	100	700
SB Leg 1	0	10	80	10	100	800

At the bottom of the dialog box are "OK" and "Cancel" buttons.

Specify percentages of traffic volumes plus the traffic volumes for a 4 leg Alternative. See the description of dialog box usage under **Primitive Command: TOOLS - Traffic - Volume**.

Primitive Command: **TOOLS - Traffic - Volume - Percent+VOL (5 legs)**



The dialog box is titled "Traffic Turn Movement Count" and has a close button in the top right corner. It features an "INPUT MODE" dropdown menu set to "Percentages of Traffic Volume". The main area is divided into "Inbound Legs" and "Outbound Legs" sections. The "Inbound Legs" section has five rows, each with a diagram of a leg and a table of percentages. The "Outbound Legs" section has five columns, each with a diagram of a leg and a table of percentages. The "Total Percent" and "Total Volume" columns are on the right. At the bottom are "OK" and "Cancel" buttons.

		Outbound Legs					Total Percent	Total Volume
		1	2	3	4	5		
Inbound Legs	1	0 U	10 L	70 S	10 R	10 R	100	500
	2	10 R	0 U	10 L	60 S	20 S	100	600
	3	70 S	10 R	0 U	10 L	10 S	100	700
	4	10 L	70 S	10 R	0 U	10 L	100	800
	5	10 L	40 S	40 S	10 R	0 U	100	900

Specify percentages of traffic volumes plus the traffic volumes for a 5 leg Alternative. See the description of dialog box usage under **Primitive Command: TOOLS - Traffic - Volume**.

Primitive Command: **TOOLS - Traffic - Volume - Percent+VOL (6 legs)**

Traffic Turn Movement Count

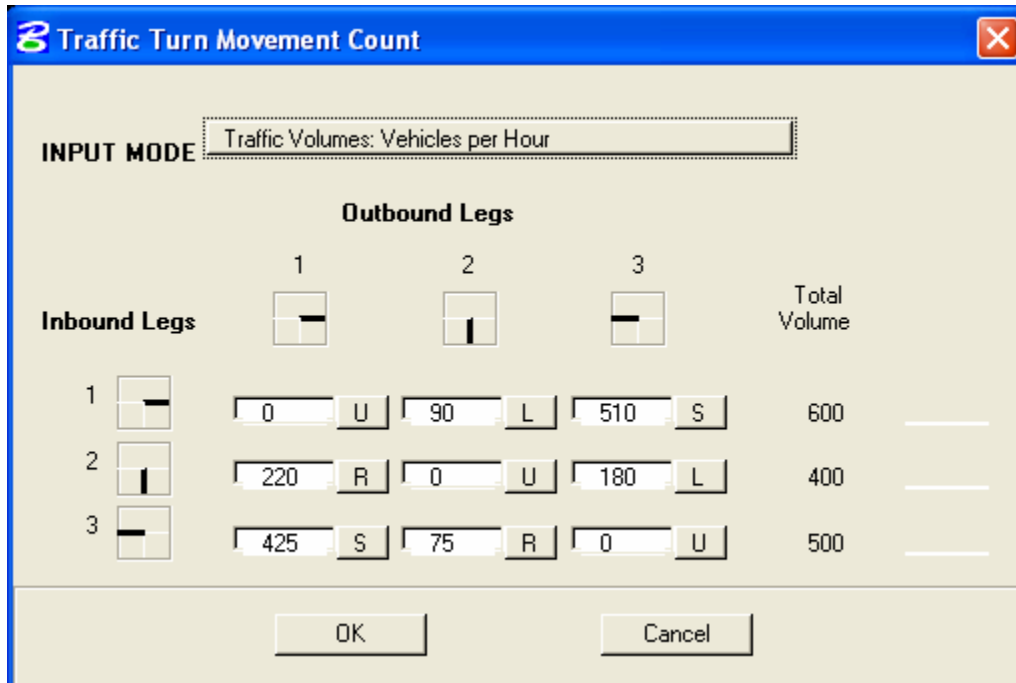
INPUT MODE: Percentages of Traffic Volume

		Outbound Legs						Total Percent	Total Volume
Inbound Legs		1	2	3	4	5	6		
1		0 U	10 L	20 S	50 S	10 R	10 R	100	400
2		10 R	0 U	10 L	10 L	50 S	20 S	100	500
3		20 S	10 R	0 U	10 L	10 L	50 S	100	600
4		50 S	10 R	10 R	0 U	10 L	20 S	100	700
5		10 L	50 S	20 R	10 R	0 U	10 L	100	800
6		10 L	10 L	50 S	20 S	10 R	0 U	100	900

OK Cancel

Specify percentages of traffic volumes plus the traffic volumes for a 6 leg Alternative. See the description of dialog box usage under **Primitive Command: TOOLS - Traffic - Volume**.

Primitive Command: TOOLS - Traffic - Volume - TMC Volume (3 legs)



The dialog box is titled "Traffic Turn Movement Count". It features a blue title bar with a close button (X) in the top right corner. Below the title bar, there is a section labeled "INPUT MODE" with a dropdown menu currently set to "Traffic Volumes: Vehicles per Hour".

The main area of the dialog is divided into two sections: "Inbound Legs" and "Outbound Legs".

Inbound Legs: This section contains three rows, numbered 1, 2, and 3. Each row has a small diagram of a traffic light showing the active movement (indicated by a horizontal bar for right-turn and a vertical bar for through/left-turn). Row 1 shows a horizontal bar, Row 2 shows a vertical bar, and Row 3 shows a horizontal bar.

Outbound Legs: This section contains three columns, numbered 1, 2, and 3. Each column has a small diagram of a traffic light showing the active movement. Column 1 shows a horizontal bar, Column 2 shows a vertical bar, and Column 3 shows a horizontal bar.

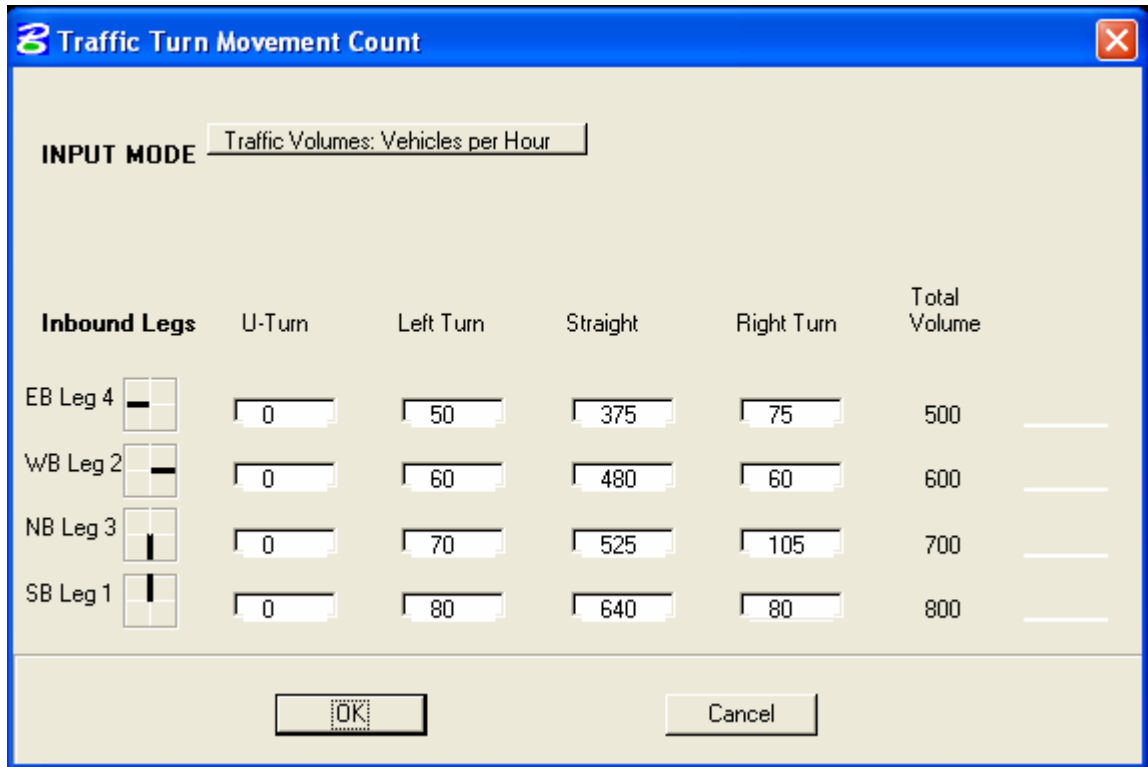
Below the diagrams, there is a table of input fields for volume and turn type. The table has 4 columns: Inbound Leg, Outbound Leg 1, Outbound Leg 2, Outbound Leg 3, and Total Volume. The data is as follows:

Inbound Legs	1	2	3	Total Volume
1	0 U	90 L	510 S	600
2	220 R	0 U	180 L	400
3	425 S	75 R	0 U	500

At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Specify turn movement count volumes for a 3 leg Alternative. See the description of dialog box usage under **Primitive Command: TOOLS - Traffic - Volume**.

Primitive Command: TOOLS - Traffic - Volume - TMC Volume (4 legs)



The dialog box is titled "Traffic Turn Movement Count" and features a close button in the top right corner. Below the title bar, there is a section labeled "INPUT MODE" with a dropdown menu currently set to "Traffic Volumes: Vehicles per Hour". The main area of the dialog contains a table for inputting traffic volumes for four inbound legs. Each leg has a small diagram of a traffic signal with a specific movement highlighted (left turn for EB Leg 4, right turn for WB Leg 2, left turn for NB Leg 3, and left turn for SB Leg 1). The table columns are "Inbound Legs", "U-Turn", "Left Turn", "Straight", "Right Turn", and "Total Volume". The "Total Volume" column has a corresponding empty box to its right for manual entry. At the bottom of the dialog are "OK" and "Cancel" buttons.

Inbound Legs	U-Turn	Left Turn	Straight	Right Turn	Total Volume	
EB Leg 4	0	50	375	75	500	
WB Leg 2	0	60	480	60	600	
NB Leg 3	0	70	525	105	700	
SB Leg 1	0	80	640	80	800	

Specify turn movement count volumes for a 4 leg Alternative. See the description of dialog box usage under **Primitive Command: TOOLS - Traffic - Volume**.

Primitive Command: **TOOLS - Traffic - Volume - TMC Volume (5 legs)**

Traffic Turn Movement Count

INPUT MODE: Traffic Volumes: Vehicles per Hour

		Outbound Legs					Total Volume
		1	2	3	4	5	
Inbound Legs							
1		0 U	50 L	350 S	50 R	50 R	500
2		60 R	0 U	60 L	360 S	120 S	600
3		490 S	70 R	0 U	70 L	70 S	700
4		80 L	560 S	80 R	0 U	80 L	800
5		90 L	360 S	360 S	90 R	0 U	900

OK Cancel

Specify turn movement count volumes for a 5 leg Alternative. See the description of dialog box usage under **Primitive Command: TOOLS - Traffic - Volume**.

Primitive Command: TOOLS - Traffic - Volume - TMC Volume (6 legs)

Traffic Turn Movement Count

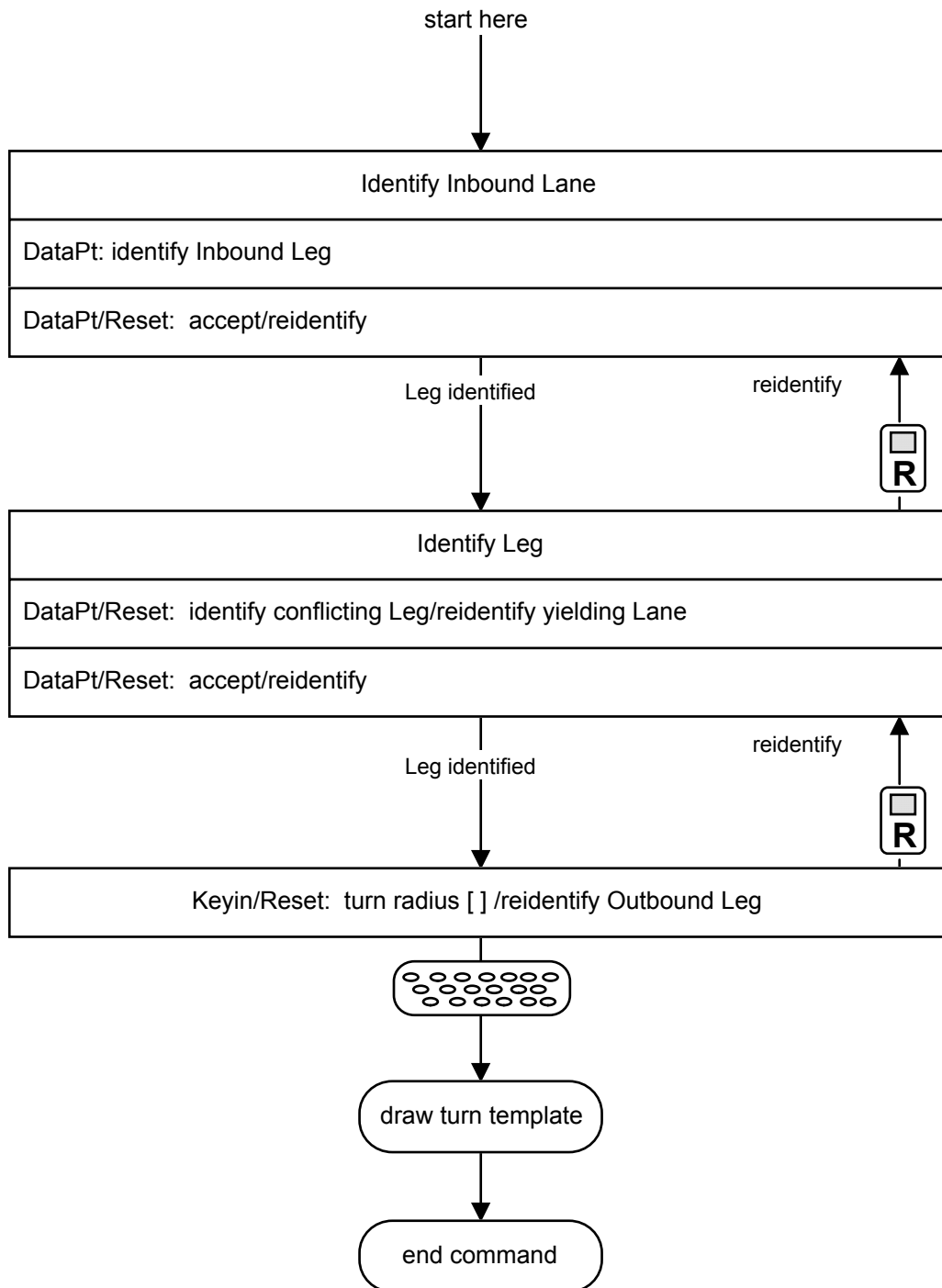
INPUT MODE Traffic Volumes: Vehicles per Hour

		Outbound Legs						Total Volume
		1	2	3	4	5	6	
Inbound Legs								
1		0 U	40 L	80 S	200 S	40 R	40 R	400
2		50 R	0 U	50 L	50 L	250 S	100 S	500
3		120 S	60 R	0 U	60 L	60 L	300 S	600
4		350 S	70 R	70 R	0 U	70 L	140 S	700
5		80 L	400 S	160 R	80 R	0 U	80 L	800
6		90 L	90 L	450 S	180 S	90 R	0 U	900

OK Cancel

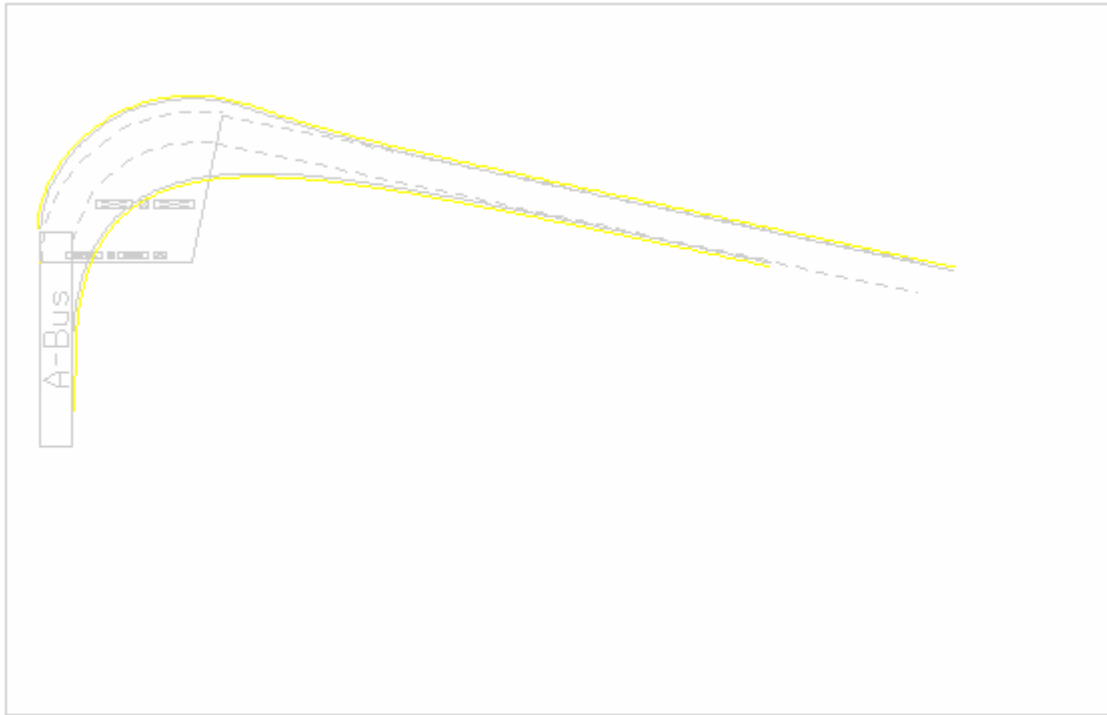
Specify turn movement count volumes for a 6 leg Alternative. See the description of dialog box usage under **Primitive Command: TOOLS - Traffic - Volume**.

Primitive Command: TOOLS - TurnTemplate

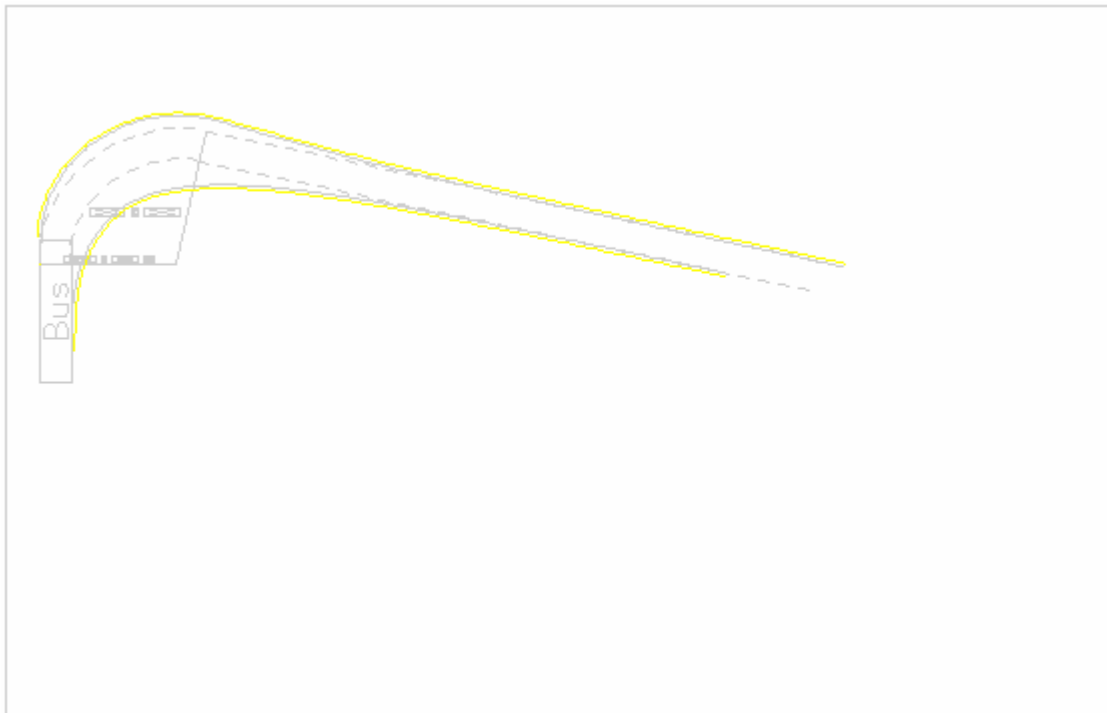


Use this diagram for the 15 **TOOLS - TurnTemplate** commands that follow. All the following vehicle turn templates were placed at the same origin with the default radius, a 1-foot safety zone, and a turn angle of 102 degrees.

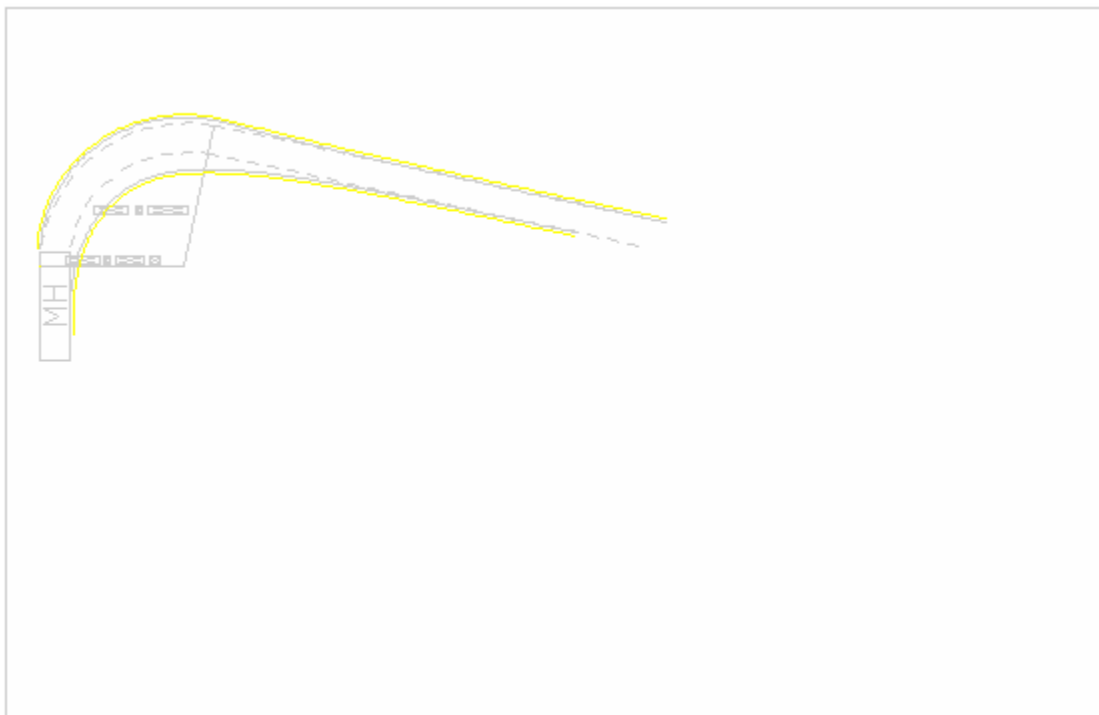
Primitive Command: TOOLS - TurnTemplate - A-Bus
Draw a turning vehicle template for an articulated bus.



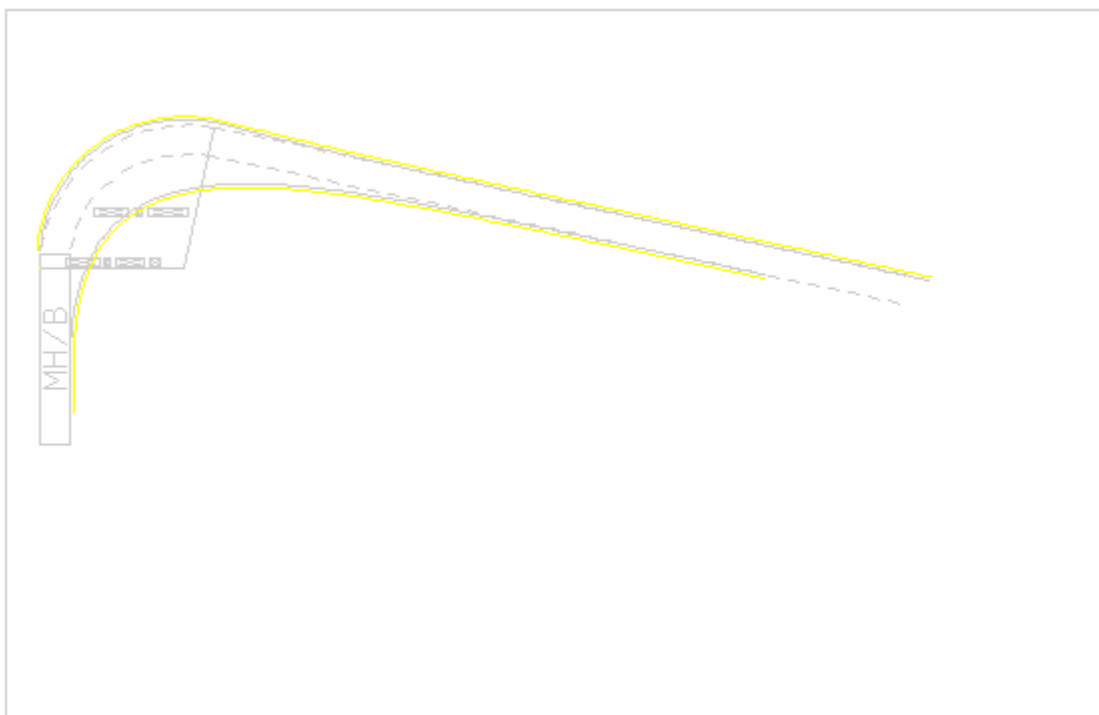
Primitive Command: TOOLS - TurnTemplate - Bus
Draw a turning vehicle template for a bus.



Primitive Command: TOOLS - TurnTemplate - MH
Draw a turning vehicle template for a motor home.

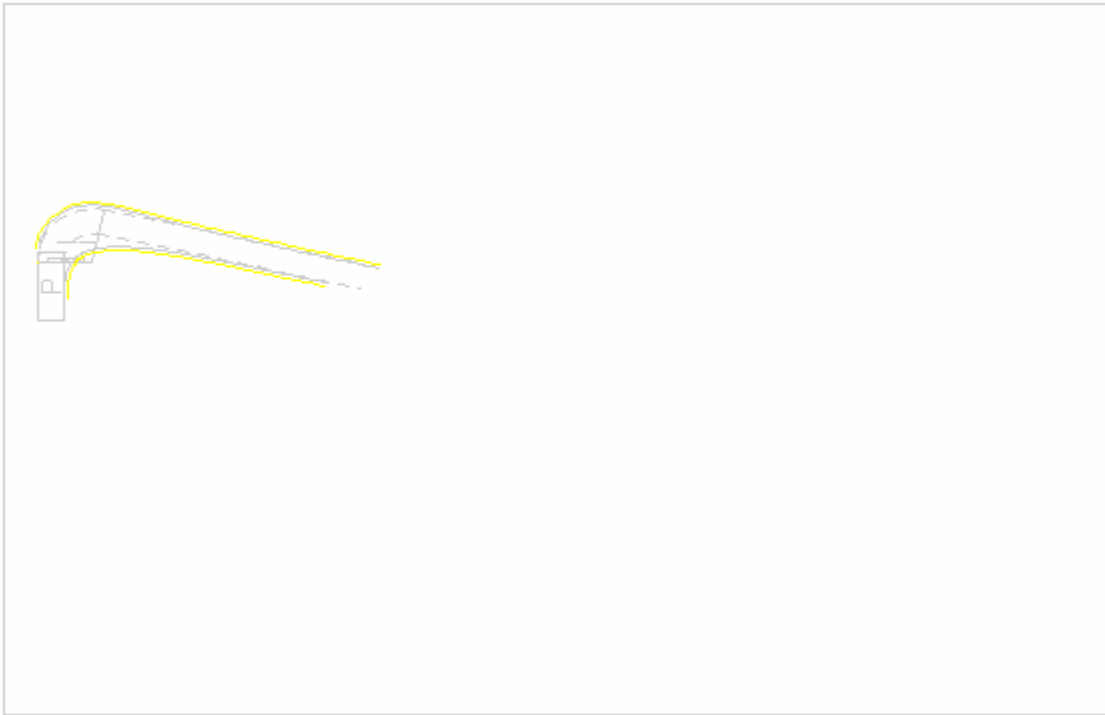


Primitive Command: TOOLS - TurnTemplate - MH/B
Draw a turning vehicle template for a motor home/bus.



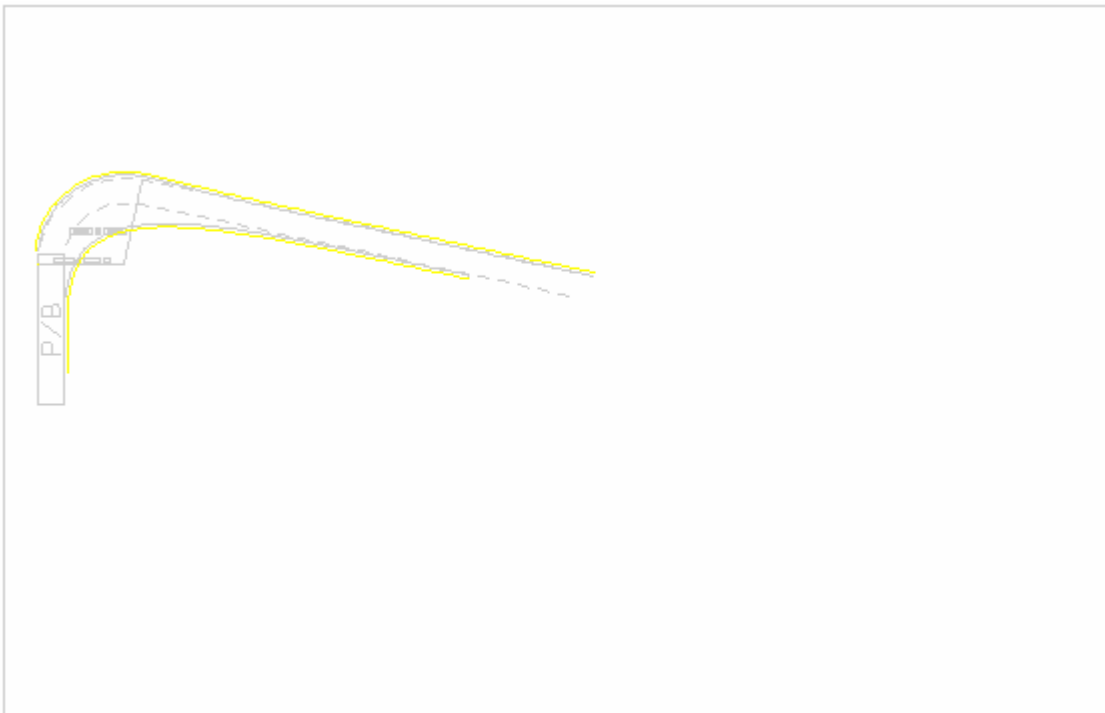
Primitive Command: TOOLS - TurnTemplate - P

Draw a turning vehicle template for a passenger vehicle.



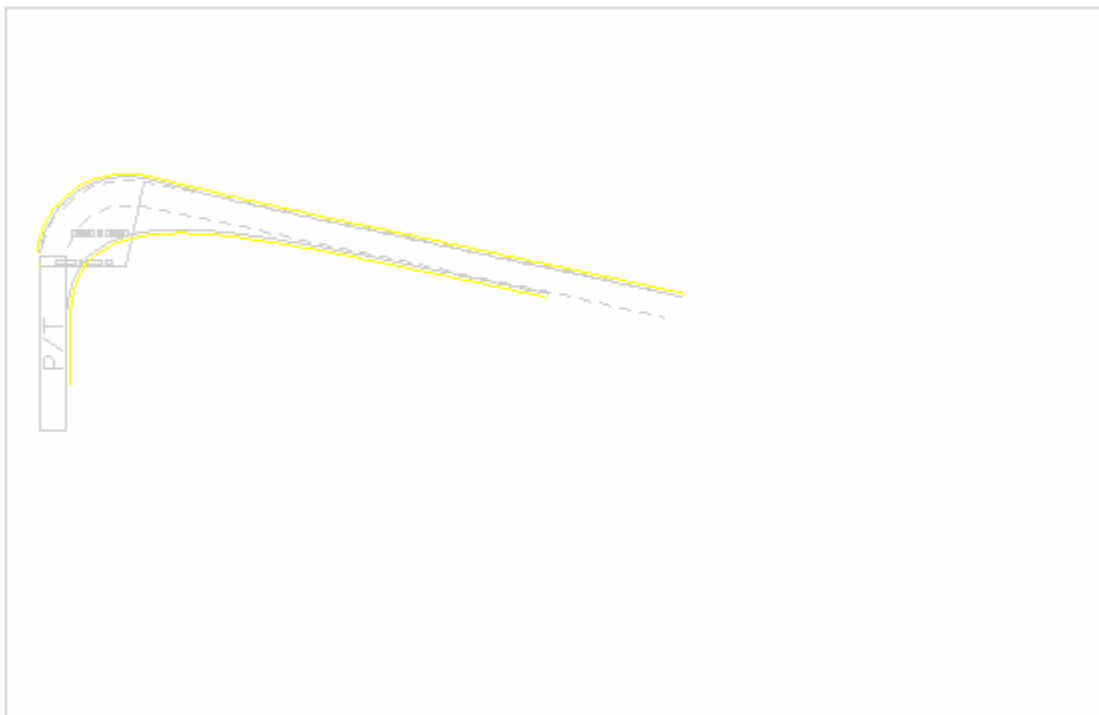
Primitive Command: TOOLS - TurnTemplate - P/B

Draw a turning vehicle template for a passenger vehicle pulling a boat.



Primitive Command: TOOLS - TurnTemplate - P/T

Draw a turning vehicle template for a passenger vehicle pulling a trailer.



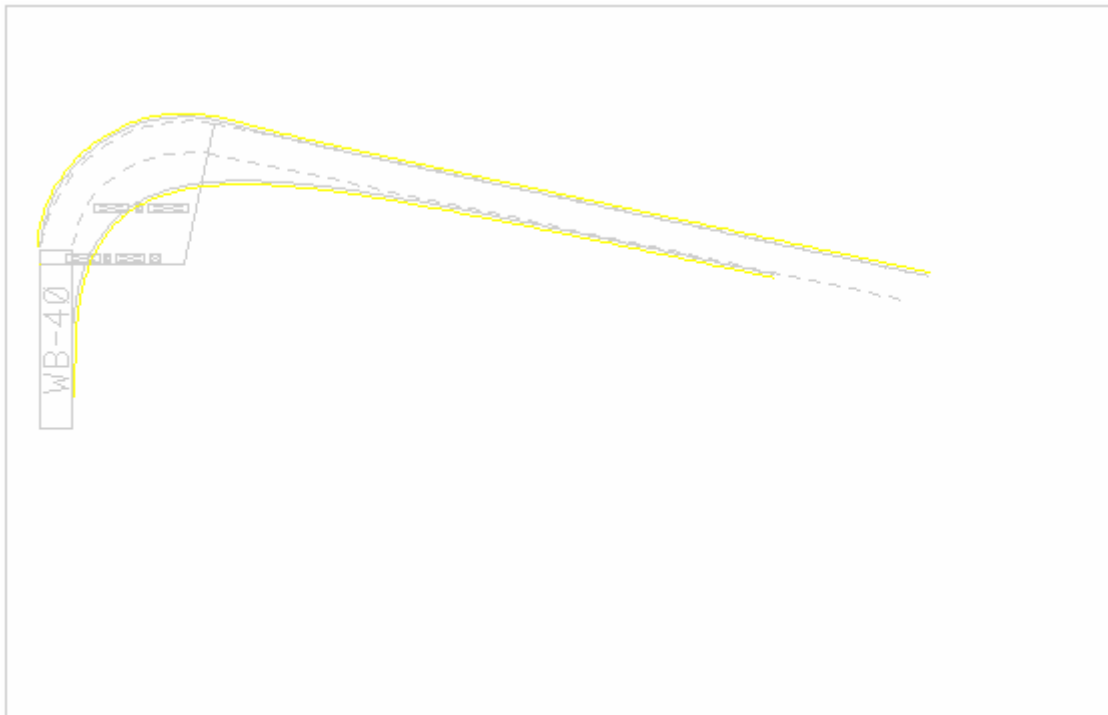
Primitive Command: TOOLS - TurnTemplate - SU

Draw a turning vehicle template for a single unit truck.



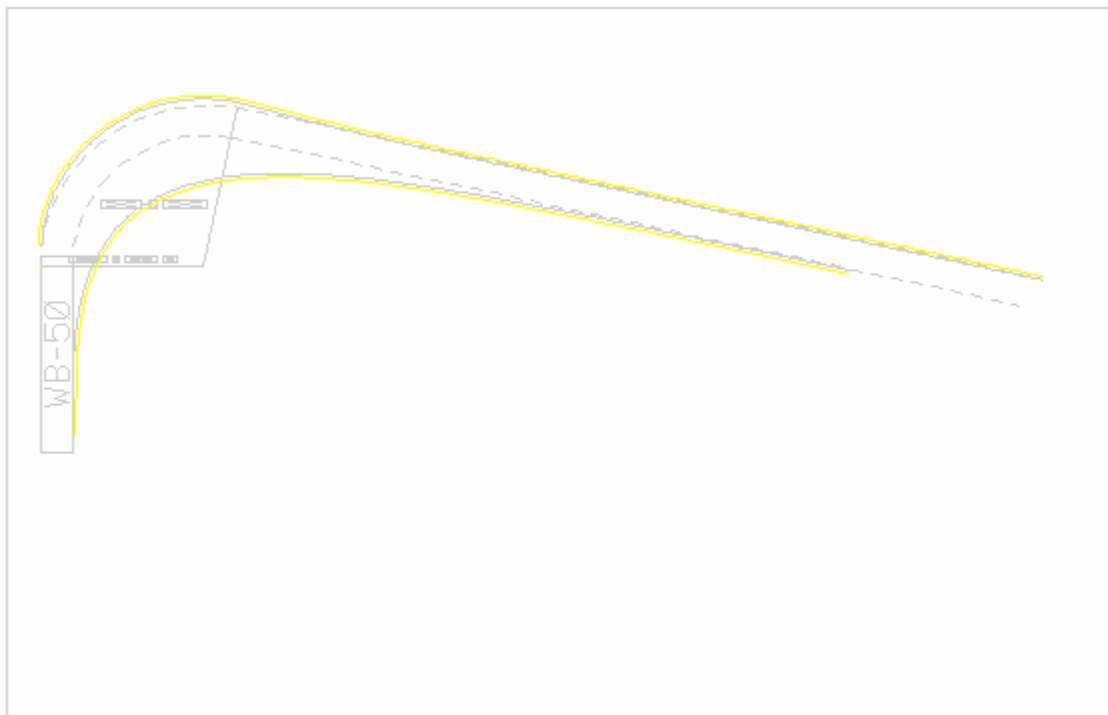
Primitive Command: TOOLS - TurnTemplate - WB-40-12

Draw a turning vehicle template for a medium tractor-semitrailer.



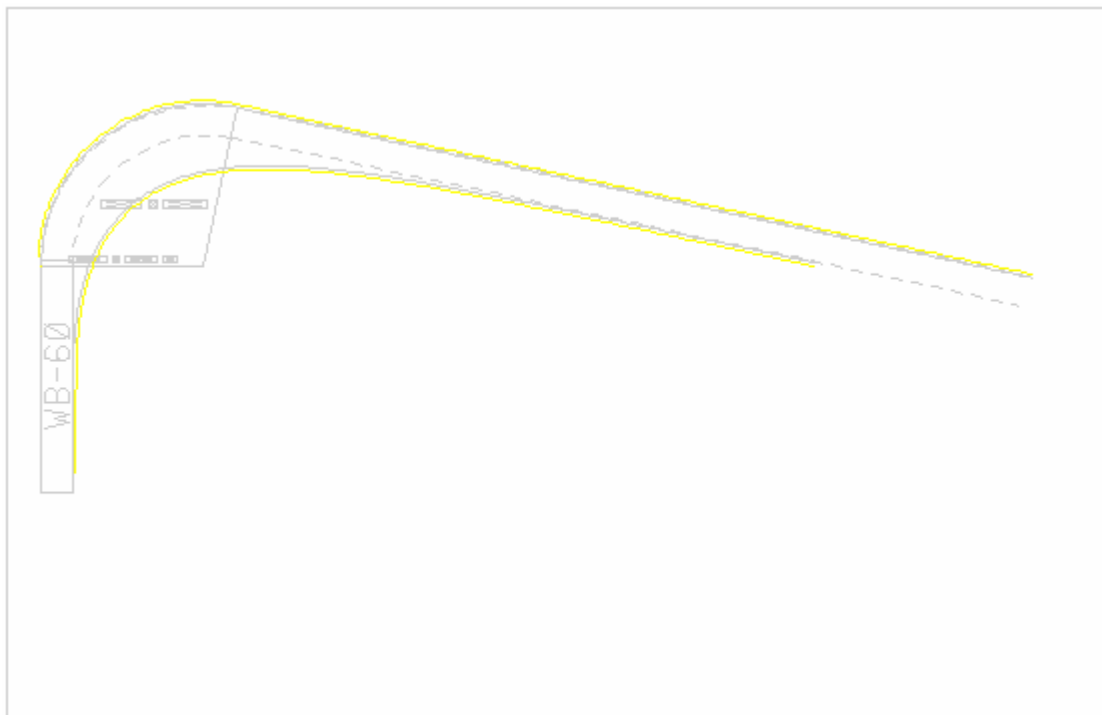
Primitive Command: TOOLS - TurnTemplate - WB-50-15

Draw a turning vehicle template for a larger tractor-semitrailer combination.



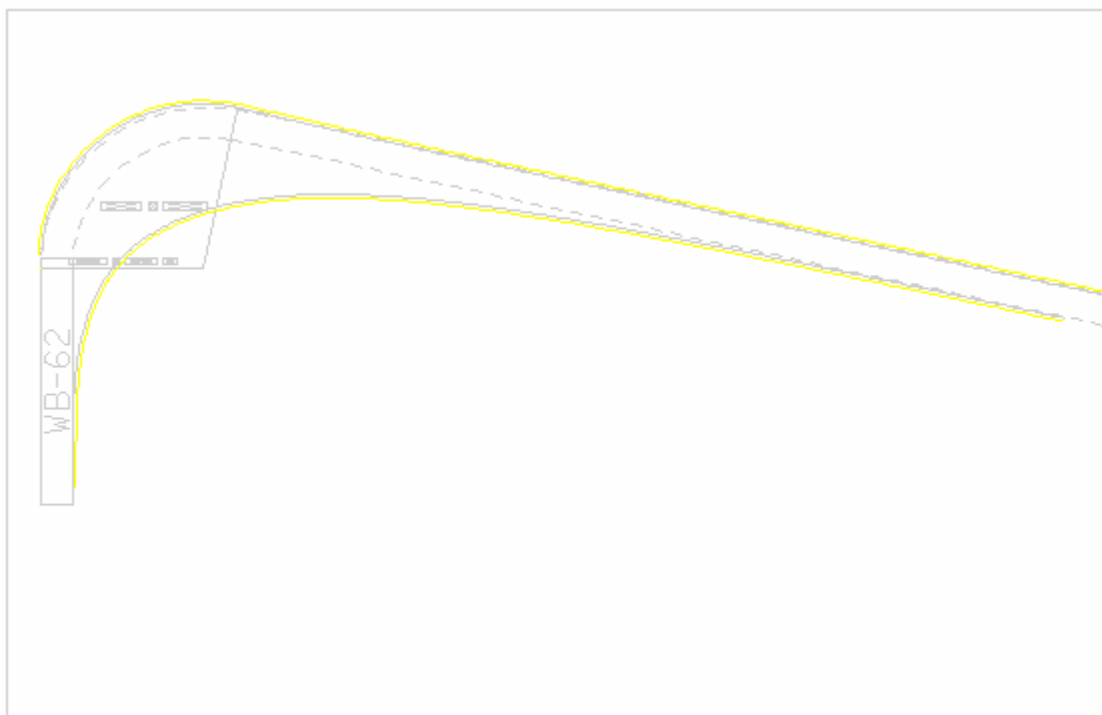
Primitive Command: TOOLS - TurnTemplate - WB-60-18

Draw a turning vehicle template for tractor-semitrailer-full trailer combinations.



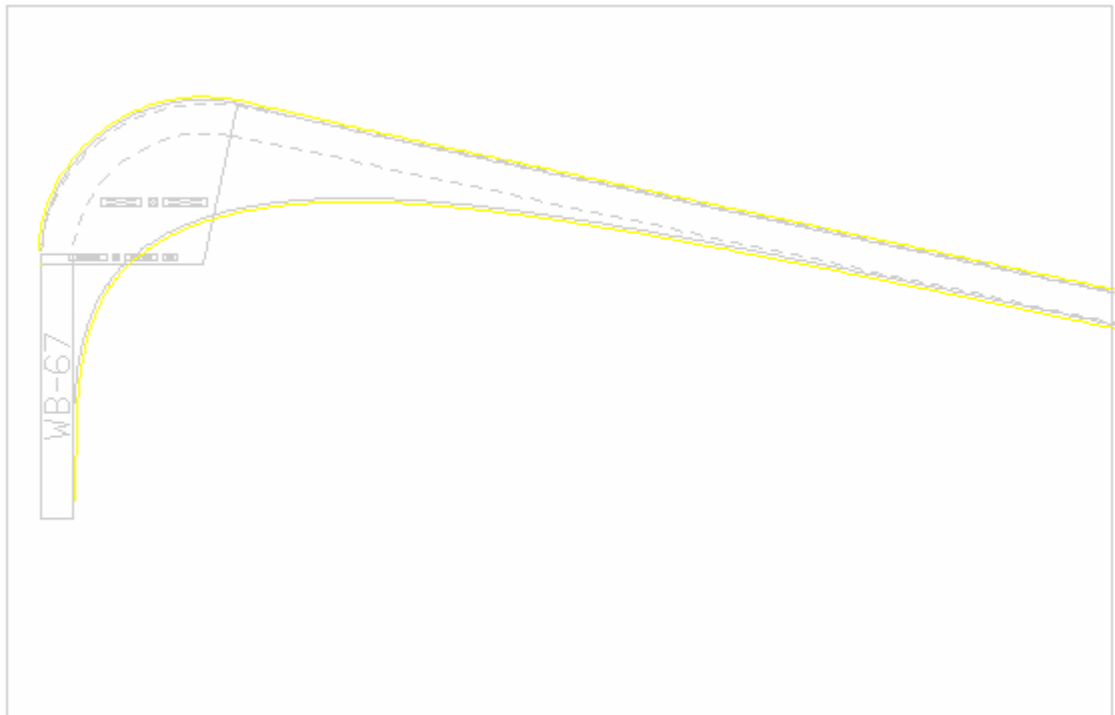
Primitive Command: TOOLS - TurnTemplate - WB-62-19

Draw a turning vehicle template for tractor-semitrailer combinations.



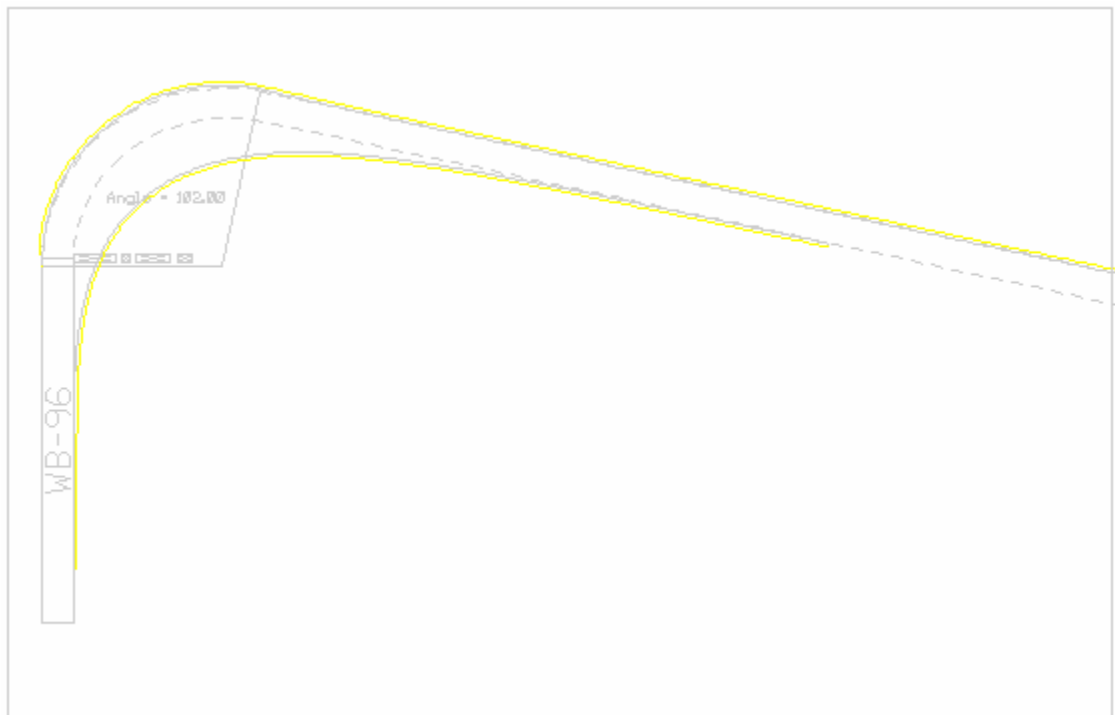
Primitive Command: TOOLS - TurnTemplate - WB-67-20

Draw a turning vehicle template for tractor-semitrailer combinations.



Primitive Command: TOOLS - TurnTemplate - WB-96-29

Draw a turning vehicle template for tractor-semitrailer-full trailer combinations.



Primitive Command: TOOLS - TurnTemplate - WB-114-35

Draw a turning vehicle template for tractor-semitrailer-full trailer combinations.



Transient Command: TOOLS - TurnTemplate - Del Graphics
Delete all of the turning vehicle template graphics.

Transient Command: Verb-Noun or Noun-Verb

This command toggles between the two methods of command processing. The **Verb-Noun** method always prompts the user to identify an Object for processing. The **Noun-Verb** method always uses the selected IGIDS Object as the default choice when identifying an Object for processing.

Transient Command: VIEW - ALTERNATIVES - ALL OFF

For all Alternatives, make all graphics invisible.

Transient Command: VIEW - ALTERNATIVES - ALL ON

For all Alternatives, make all graphics visible.

Transient Command: VIEW - ALTERNATIVES - CURRENT OFF

For the selected Alternative, make all graphics invisible.

Transient Command: VIEW - ALTERNATIVES - CURRENT ON

For the selected Alternative, make all graphics visible.

Transient Command: VIEW - LANE - CURRENT OFF

For the selected Alternative, make all Inbound Lanes and Outbound Lanes invisible.

Transient Command: VIEW - LANE - CURRENT ON

For the selected Alternative, make all Inbound Lanes and Outbound Lanes visible.

Transient Command: VIEW - LEG CNTRLIN - CURRENT OFF

For the selected Alternative, make all Centerline Segs invisible.

Transient Command: VIEW - LEG CNTRLIN - CURRENT ON

For the selected Alternative, make all Centerline Segs visible.

Transient Command: VIEW - TEXT - CURRENT OFF

For the selected Alternative, make all Text invisible.

Transient Command: VIEW - TEXT - CURRENT ON

For the selected Alternative, make all Text visible.

Transient Command: VIEW - TRAF CONTROL - CURRENT OFF

For the selected Alternative, make all of the traffic control device graphics invisible.

Transient Command: VIEW - TRAF CONTROL - CURRENT ON

For the selected Alternative, make all of the traffic control device graphics visible.

Transient Command: Yes

In reply to a prompt requesting a "yes" or "no" response, send "yes" to IGIDS. This is the same as entering "yes" through the keyboard.

Transient Command: [default]

In response to a prompt, send the default value to IGIDS. When a default value is acceptable, it will be shown in the prompt, enclosed by square brackets. The default value may also be sent through the keyboard by pressing only the return key. For example, the prompt "Keyin: Lane width[12]" indicates that IGIDS will use 12 as the default value for lane width.