ENF0000AN010

MT Decoding NMEA Messages

Application Note

Version: 1.02

NOVATEL WIRELESS

02 December, 2013

General

TERMS OF USE OF NEW MATERIALS - PLEASE READ CAREFULLY

From time to time, Novatel Wireless M2M, in its sole discretion, may make available for download on its website (www.novatelwireless.com), or may transmit via mail or email, updates or upgrades to, or new releases of, the firmware, software or documentation for its products (collectively, 'New Materials'). Use of such New Materials is subject to the terms and conditions set forth below, and may be subject to additional terms and conditions as set forth in Novatel Wireless M2M's Technical Support Policy (posted on its website) and/or any written agreement between the user and Novatel Wireless M2M.

All New Materials are provided AS IS. Novatel Wireless M2M makes no warranty or representation with respect to the merchantability, suitability, functionality, accuracy or completeness of any such New Materials. The user of such New Materials assumes all risk (known or unknown) of such use. Novatel Wireless M2M reserves all rights in such New Materials. The user shall have only a revocable and limited license to use such New Materials in connection with the products for which they are intended. Distribution or modification of any New Materials without Novatel Wireless M2M's consent is strictly prohibited.

IN NO EVENT WILL Novatel Wireless M2M BE RESPONSIBLE FOR ANY INCIDENTAL, INDIRECT, CONSEQUENTIAL OR SPECIAL DAMAGES AS A RESULT OF THE USE OF ANY NEW MATERIALS. Novatel Wireless M2M'S MAXIMUM LIABILITY FOR ANY CLAIM BASED ON THE NEW MATERIALS SHALL NOT EXCEED FIFTY U.S. DOLLARS ($50).

Version Verification

Please ensure you have the latest version of this document by downloading it from www.novatelwireless.com

Copyright

251 Renner Pkwy

Richardson, TX 75080 USA

Phone: (972) 633-4400

Fax: (972) 633-4444

Email: info@nvtl.com

www.novatelwireless.com

Table of Contents

1

Introduction

Objective

Objective

This document discusses how to plot coordinates provided by the MT's GPS receiver when configured for NMEA or reporting binary to a server. While this example focuses on GPRMC and Binary reporting, you can use any NMEA sentence that provides latitude and longitude for this example.

1

Message Decoding

Overview

MT series modems have GPS receivers that can report current location using the NMEA protocol. Data can be captured locally via serial port, local UDP, or remote UDP. In the following examples, the Novatel Wireless, Inc. address was used for mapping coordinates:

Novatel Wireless, Inc.

251 Renner Pkwy

Richardson, TX 75080

Sample Data

$GPRMC Sentence Received:

```
$GPRMC,165717.00,A,3259.816776,N,09642.858868,W,0.0,0.0,070108,3.5,
W,A*30
```

This message reports the time, date, position, course, and speed from the receiver's navigation solution. See Table 1 for a description of RMC.

ASCII Server Message Received (param2 =4102):

```
<00><04><02><00> 1234 MTGPSTEST
$GPRMC,165717.00,A,3259.816776,N,09642.858868,W,0.0,0.0,070108,3.5,
W,A*30
```

This message reports the following from the receiver's navigation solution:

- UDPAPI Header
- Param1
- MDMID
- Time
- Date
- Position
- Course
- Speed

See Table 2 for a description of ASCII Payload.

Binary Server Message Received (param2 =3079):

```
<00><05><02><00><00><00><04><D2><20><20><20><20><20><20><20><20>
<20><20><20><4D><54><47><50><53><54><45><53><54><20><31><BD><A8><FF>
<6C><DC><96>
```

This message reports the following from the receiver's navigation solution:

- UDPAPI Header
- Param1
- MDMID
- Position

See Table 2 for a description of Binary Payload.

Since we captured the reports above from an MT in a stationary location, we consider the values shown for “Speed Over Ground” and “Course Over Ground” as invalid. The unit must be in mostion for these values to be valid.

A Google search provides a wealth of information on the NMEA protocol.
All latitude/longitude and course over ground data is in WGS84. In order to use the latitude/longitude data directly, the host/mapping application must be using WGS84 or a translation must be performed prior to plotting the location.

$GPRMC Sentence Decode

Name	Example	Units	Description
Message ID	$GPRMC		RMC protocol header
UTC Position	165717.00		hh mm ss.sss
Status	A		A = data valid or V = data not valid or 9 = last known GPS location
Latitude	3259.816776		dd mm.mmmm
N/S Indicator	N		N = north or S = south
Longitude	09642.858868		ddd mm.mmmm
E/W Indicator	W		E = east or W = west
Speed Over Ground	0.0	knots	
Course Over Ground	0	degrees	True
Date	070108		dd mm yy
Magnetic Variation*	3.5	degrees	
	W		E = east or W = west
Position Mode Indicator*	A		A = Autonomous
Checksum	*30		
<CR><LF>			End of message termination
*Some receivers do not support magnetic declination			
*Some receivers do not support Position Mode Indicator			

Table 1-1 RMC

ASCII Server Message Decode

Name	Example	Units	Description
UDPAPI Header	0420		UDPAPI Header ASCII Message
Param1	1234		Param1
MDMID	MTGPSTEST		Modem ID
Message ID	$GPRMC		RMC protocol header
UTC Position	165717.00		hh mm ss.sss
Status	A		A = data valid or V = data not valid or 9 = last known GPS location
Latitude	3259.816776		dd mm.mmmm
N/S Indicator	N		N = north or S = south
Longitude	09642.858868		ddd mm.mmmm
E/W Indicator	W		E = east or W = west
Speed Over Ground	0.0	knots	
Course Over Ground	0	degrees	True

Name	Example	Units	Description
Date	070108		dd mm yy
Magnetic Variation*	3.5	degrees	
	W		E = east or W = west
Position Mode Indicator*	A		A = Autonomous
Checksum	*30		
<CR><LF>			End of message termination
*Some receivers do not support magnetic declination			
*Some receivers do not support Position Mode Indicator			

Table 1-2 ASCII Payload

Binary Server Message Decode

Name	Data	Decode	Description
UDPAPI Header	<00><05><02><00>	0520	UDPAPI Header Binary Message
Param1	<00><00><04><D2>	1234	Param1 (4 bytes)
MDMID	<20><20><20><20><20><20><20><20><20><20><20><20><4d><54><47><50><53><54><45><53><54><20>	MTGPSTEST	Modem ID (22 bytes)
Latitude	<31><BD><A8>	31BDA8	Raw latitude (3 bytes)
Longitude	<FF><6C><DC><96>	FF6CDC96	Raw Longitude (4 bytes)

Table 1-3 Binary Payload

1

Translating Between Formats

Determining The North/South & East/West Indicator

In the absence of a separate North/South, East/West indicator, the sign bit (the -/+ sign of the number) will indicate the globe quadrant for the reported position.

The Earth is divided into 360 degrees and then again in half by Latitude and Longitude. The degree of Latitude (North and South) and Longitude (East and West) ranges as follows:

Latitude N = 90

Latitude S = -90

Longitude E = 180

Longitude W = -180

The following diagram indicates how to determine negative latitudes and longitudes.

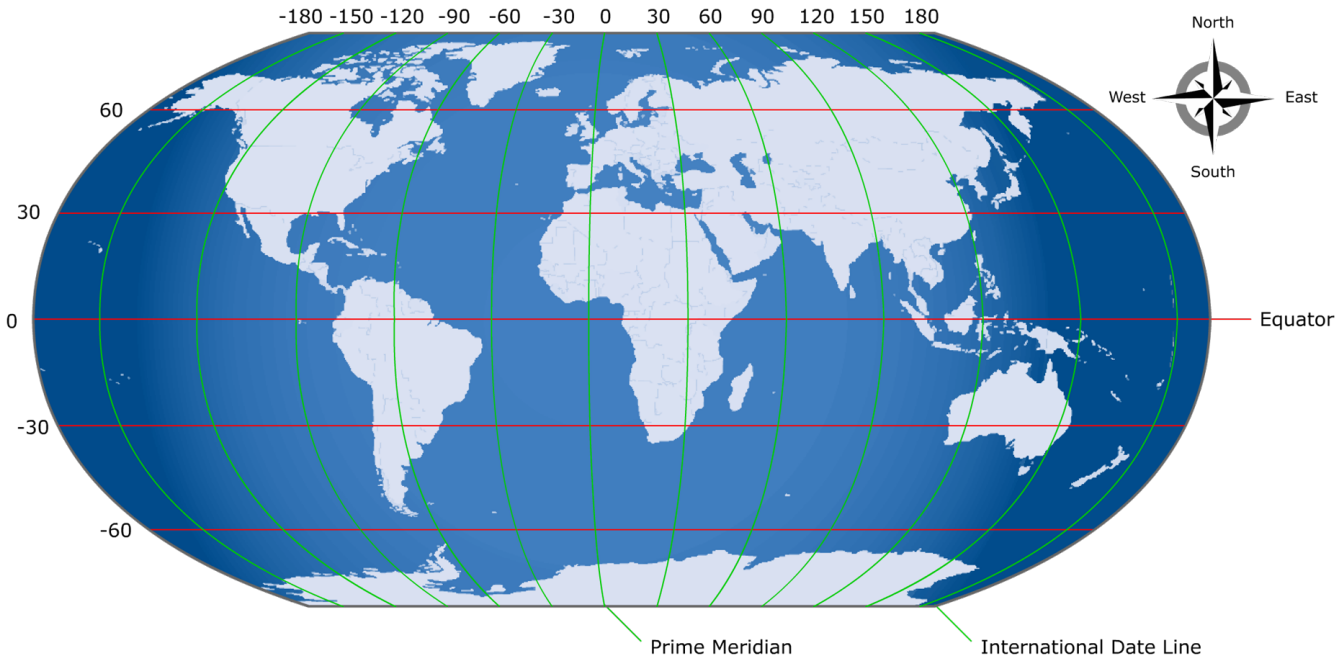


Figure 1-1 Latitude and Longitude Negative and Positive Determination

Defining Latitude/Longitude Formats

There are several formats used to represent Latitude and Longitude, including:

Degrees Minutes Decimal Seconds (DMS):

Latitude: DD MM SS.SSSS

Longitude: DDD MM SS.SSSS

Example:

Latitude: 40°44′54″N

Longitude: -73°59′10″W

Degrees (40°): The first whole number

Minutes(44′): A minute indicates 1/60 of one degree. Each of the Earth's 360 degrees are divided into 60 sections, or minutes.

Seconds(54″): A second indicates 1/60 of one minute. Each of the Earth's 360 degrees are divided into 60 sections, or minutes, and each minute is divided into 60 sections, or seconds.

Decimal Degrees (DegDec):

Latitude: DD.DDDD

Longitude: DDD.DDDD

Example:

Latitude: 40.7484°N

Longitude: -73.9847°W

Degrees Minutes (MinDec):

Latitude: DD MM.MMMM

Longitude: DDD MM.MMMM

Example:

Latitude: 40°44.7717N

Longitude: -73°56.93172W

Converting Latitude And Longitude Formats

Converting DMS To Decimal Degree

When you have a DMS coordinate such as N40°44′54″, use the following method to convert it to a Decimal Degree:

1. Calculate the number of seconds: 44′54″ = (44*60 + 54) = 2694 seconds
2. Get the fractional part by dividing the total number of seconds by 3600:
   2694 / 3600 = 0.7484
3. Add the number of degrees to this fractional number:
   40+0.7484 = 40.7484

If your DMS coordinates are South or West, change the sign from (+) positive to (-) negative.

Converting MinDec To Decimal Degrees

When you have a MinDec coordinate such as 40°44.7717N, 73°56.93172W, use the following method to convert it to a Decimal Degree (for example for 73°56.93172W):

The DecimalDegree is the decimal minutes divided by 60 (56.93172/60 = 0.948862)

Add the Degree (73) to the Decimal Degree (73 + 0.948862 = 73.948862)

If your coordinates are South or West, change the sign from (+) positive to (-) negative.

Converting Decimal Degrees To DMS

When you have a decimal coordinate such as -87.728055, you need to determine whether it is Latitude or Longitude in order to convert it. In this case, it is a Longitudinal coordinate. Use the following method to convert it to DMS:

Subtract the whole number of the coordinate (leaving the fractional part). The whole number is the number of degrees:
87.728055 = 87 degrees.

Multiply the fractional part of degrees by 60. This will produce a number of minutes. Use the whole number of the coordinate (leaving the fractional part).
0.728055 x 60 = 43.6833 = 43 minutes.

Multiply the fractional part of the number of minutes by 60 to produce a number of seconds.
0.6833 x 60 = 40.998 = 41 seconds.
It is possible to count this as 40 seconds by rounding the decimal to 41 (or keep the entire whole number).

Depending on whether the original number was latitudinal or longitudinal and on the sign of the number, add the N/S/E/W indicator.

Novatel Wireless M2M Binary Reporting

Converting From NMEA To Binary Hex

Before you can transmit the latitude and longitude via Binary messages, you must convert it to hex data. Use the following formula:

First, remove extra precision.

RMC Latitude = 3259.816776 N

Latitude = 3259.816

RMC Longitude = 09642.858868 W

Longitude = 09642.858

Now remove the decimal.

Latitude = 3259.816 * 1000

Latitude = 3259816

Longitude = 09642.858 * 1000

Longitude = 09642858

Add the sign bit (if necessary) and then convert to hex.

If latitude direction is South, then:
Latitude = Latitude * -1

Latitude = 3259816 (Decimal)

Latitude = 31BDA8 (Hex)

If longitude direction is West, then:

Longitude = Longitude * -1

Longitude = -09642858 (Decimal)

Longitude = FF6CDC96 (Hex)

Converting From Binary HEX

Latitude transmits as 3 bytes (big endian) format. Longitude transmits as 4 bytes big endian format.

Converting back from HEX to Decimal can be more complicated. Because of the word size, not all conversion routines can correctly convert the sign bit. A more reliable method is to remove the sign bit, perform the conversion, and reapply the sign bit.

Some languages do not allow a –0 for degrees. If you must store the sign bit as part of the latitude and longitude, then we recommend that you store and pass the latitude and longitude as decimal degrees.

The following is based on an integer of 32 bits. If the integer you use is not 32 bits, then you may need to use different algorithms.

Converting Latitude

Latitude = 3259816

Latitude = 31BDA8

Latitude is transmitted as 3 bytes (big endian) format. Latitude = 31BDA8 is transmitted as 31,BD,A8.

When you remove the upper word, you will need to implement a filter to determine whether the latitude is negative.

latitude(2) = 31

latitude(1) = BD

latitude(0) = A8

Test the most significant bit (MSB) of the first byte to see if the latitude is negative.

This test will work accurately up to 84 degrees. For degrees above 84, you will either need NMEA data or to perform other filtering.

How To Convert Novatel Wireless M2M Hex GPS Latitudes & Longitudes

	Latitude Example 1 - Hex value = 31BDA8						
1.	Move 24 bits to low order bytes of 32-bit buffer	00	31	BD	A8		
2.	If most significant bit of most significant byte (31) is 1, prepend FF	00	31	BD	A8	=	3259816
3.	Divide by 100,000					=	32.59816
4.	Degrees = Integer portion					=	32
5.	Fractional Minutes					=	.59816
6.	Fractional Degrees = (Fractional Minutes / .6)					=	.996933
7.	Result (Degrees + Fractional Degrees)					=	32.996933

	Latitude Example 2 - Hex value = E3B700						
1.	Move 24 bits to low order bytes of 32-bit buffer	00	E3	B7	00		
2.	IF most significant bit of most significant byte (E3) is 1, prepend FF	FF	E3	B7	00	=	-1853696
3.	Divide by 100,000					=	-18.53696
4.	Degrees = Integer portion					=	-18
5.	Fractional Minutes					=	-0.53696
6.	Fractional Degrees = (Fractional Minutes / .6)					=	-0.894933
7.	Result (Degrees + Fractional Degrees)					=	-18.894933

	Longitude Example 1 - Hex value = FF6CDC96						
2.	32-Bit Buffer	FF	6C	DC	96	=	-9642858
3.	Divide by 100,000					=	-96.42858
4.	Degrees = Integer portion					=	-96
5.	Fractional Minutes					=	-0.42858
6.	Fractional Degrees = (Fractional Minutes / .6)					=	-0.714300
7.	Result (Degrees + Fractional Degrees)					=	-96.714300

C Example Of Conversion

```
If latitude(2) && 0x80
{
    \* must be negative
    latitude(3) = 0xFF
}
else
{
    \* must be positive
    latitude(3) = 0x00
}
```

Now rebuild the raw integer

latTempInteger = latitude(3)

latTempInteger = latTempInteger << 8

latTempInteger + = latitude(2)

latTempInteger = latTempInteger << 8

latTempInteger + = latitude(1)

latTempInteger = latTempInteger << 8

latTempInteger + = Latitude(0)

latTempInteger should now equal 3259816

Grab the sign bit:

latitudeSign = sign(latTempInteger)

Remove the sign bit:

latTempInteger = absolute(latTempInteger)

To get the degrees, divide by 100000 and truncate after the decimal point.

To get the decimal Minutes, subtract degrees * 100000 from latTempInteger and divide the results by 1000.

latitudeDegrees = integer (latTempInteger / 100000)

latitudeDecimalMinutes = (latTempInteger - (latitudeDegrees * 100000)) / 1000

Now all the elements are separate; you can use them to perform any other necessary conversions.

latitudeSign

latitudeDegrees

latitudeDecimalMinutes

Converting Longitude

The longitude is transmitted as 4 bytes (big endian) format.

Longitude = -09642858

Longitude = FF6CDC96

The longitude is transmitted as 4 bytes (big endian) format.

Longitude = FF6CDC96 is transmitted as FF,6C,DC,96

longitude (3) = FF

longitude (2) = 6C

longitude (1) = DC

longitude (0) = 96

Now rebuild the raw integer

longTempInteger = longitude (3)

longTempInteger = longTempInteger << 8

longTempInteger + = longitude (2)

longTempInteger = longTempInteger << 8

longTempInteger + = longitude (1)

longTempInteger = longTempInteger << 8

longTempInteger + = longitude (0)

longTempInteger should now equal -09642858

Grab the sign bit:

longitudeSign = sign(longTempInteger)

Remove the sign bit:

longTempInteger = absolute(longTempInteger)

To get the degrees, just divide by 100000 and truncate after the decimal point.

To get the decimal Minutes, just subtract degrees * 100000 from longTempInteger and divide the results by 1000.

longitudeDegrees = integer (longTempInteger / 100000)

longitudeDecimalMinutes = (longTempInteger - (longitudeDegrees * 100000)) / 1000

Now all the elements are separate; you can use them to perform any other necessary conversions.

longitudeSign

longitudeDegrees

longitudeDecimalMinutes

Examples

Using An Internet Service – Maps.Google.com

There are several services on the Internet available to display a map using the coordinates provided by the $GPRMC message. http://maps.google.com can accept many different formats and is an easy way to test your conversion routines.

Just replace <LATD> and <LONGD> with the latitude and longitude you want to use.

Example:

Latitude = 32 59.816

Longitude = -096 42.858

Using DeLorme's "Street Atlas USA Versions 6.0 To 8.0"

Street Atlas has an option to "Import Lat/Long File...". By using this program to plot the data, there is no need to convert the latitude and longitude coordinates extracted from the $GPRMC sentence.

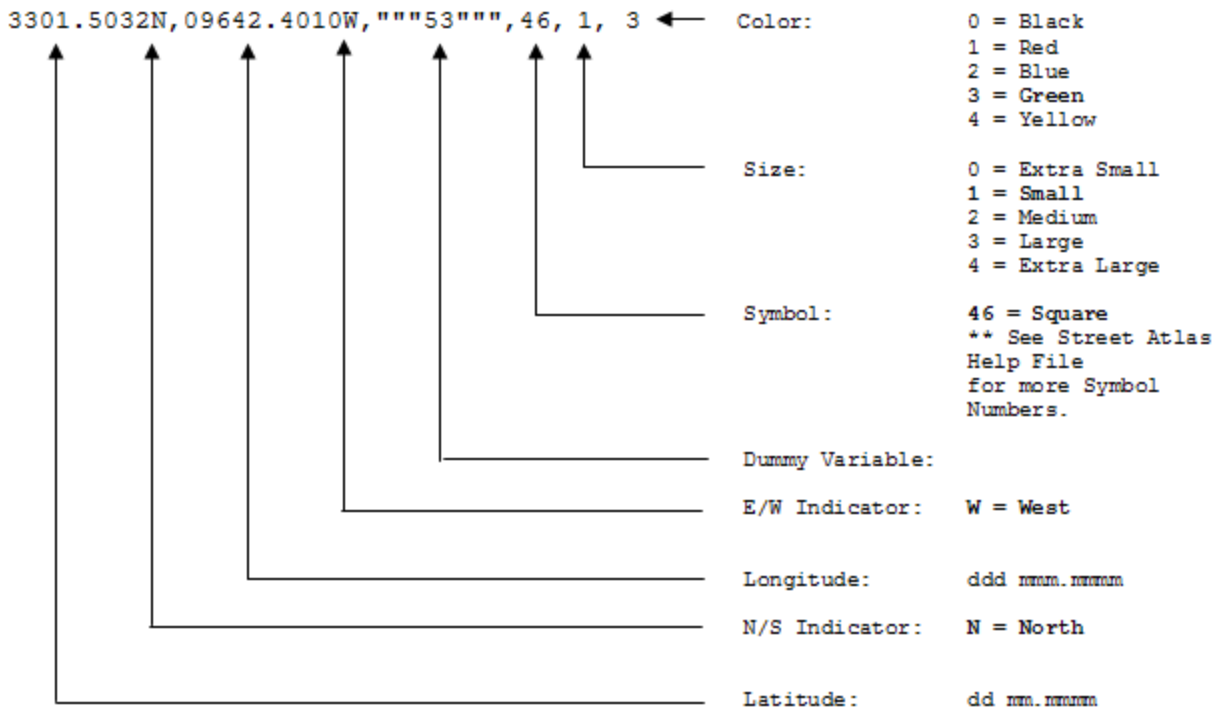
To use this function, create a text file containing the following comma delimited data:

```
Latitude N/S Indicator, Longitude E/W Indicator, Dummy Field,
Symbol=Square, Size, Color
```

Save this file with the *.TXT extension. In the Street Atlas program, click on the "File" menu and then on "Import Lat/Long File...". A dialog box will appear; choose the *.TXT file that you created. Street Atlas will plot the coordinate as shown in figure-2.

You should configure Street Atlas's map screen to display the approximate location of where you will plot the coordinate. Street Atlas will not center the view on the plotted coordinate. All values for the plot.txt file were extracted from the example $GPRMC message as described above.

Example:



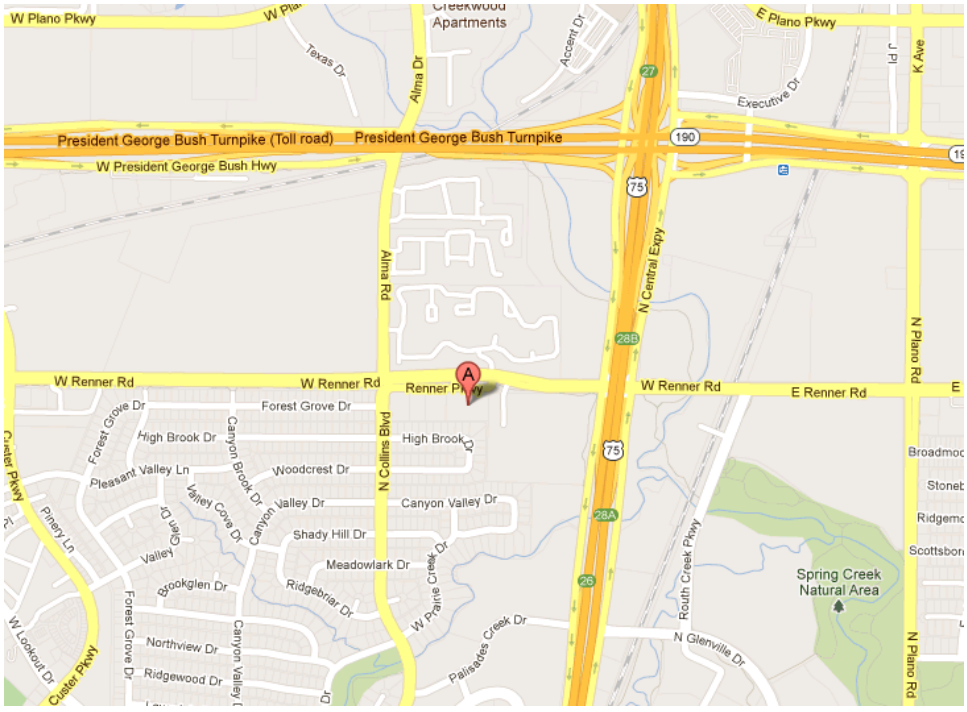


Figure 1-2 Street Atlas Plot